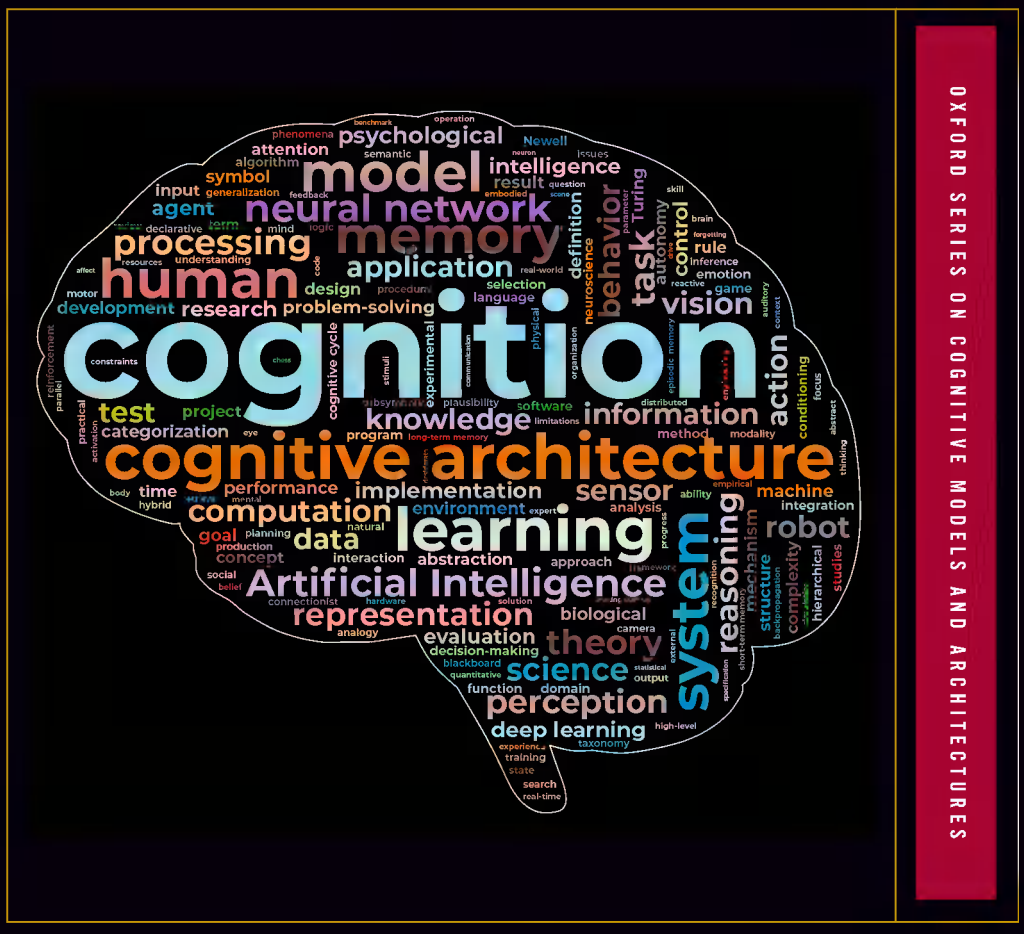

THE COMPUTATIONAL EVOLUTION OF COGNITIVE ARCHITECTURES

IULIIA KOTSERUBA AND
JOHN K. TSOTSOS

OXFORD

The Computational Evolution of Cognitive Architectures

OXFORD SERIES ON COGNITIVE MODELS AND ARCHITECTURES

Series Editor
Frank E. Ritter

Series Board
Rich Carlson
Gary Cottrell
Robert L. Goldstone
Eva Hudlicka
William G. Kennedy
Pat Langley
Robert St. Amant

Integrated Models of Cognitive Systems
Edited by Wayne D. Gray

In Order to Learn: How the Sequence of Topics Influences Learning
Edited by Frank E. Ritter, Joseph Nerb, Erno Lehtinen, and Timothy O'Shea

How Can the Human Mind Occur in the Physical Universe?
By John R. Anderson

Principles of Synthetic Intelligence PSI: An Architecture of Motivated Cognition
By Joscha Bach

The Multitasking Mind
By David D. Salvucci and Niels A. Taatgen

How to Build a Brain: A Neural Architecture for Biological Cognition
By Chris Eliasmith

Minding Norms: Mechanisms and Dynamics of Social Order in Agent Societies
Edited by Rosaria Conte, Giulia Andrighetto, and Marco Campennì

Social Emotions in Nature and Artifact
Edited by Jonathan Gratch and Stacy Marsella

Anatomy of the Mind: Exploring Psychological Mechanisms and Processes with the Clarion Cognitive Architecture
By Ron Sun

Exploring Robotic Minds: Actions, Symbols, and Consciousness as Self-Organizing Dynamic Phenomena
By Jun Tani

Brain–Mind: From Neurons to Consciousness and Creativity
By Paul Thagard

Mind–Society: From Brains to Social Sciences and Professions
By Paul Thagard

Natural Philosophy: From Social Brains to Knowledge, Reality, Morality, and Beauty
By Paul Thagard

Computational Models of Reading: A Handbook
By Erik D. Reichle

The Computational Evolution of Cognitive Architectures

Iuliia Kotseruba
and
John K. Tsotsos

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP,
United Kingdom

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide. Oxford is a registered trade mark of
Oxford University Press in the UK and in certain other countries.

Published in the United States of America by Oxford University Press
198 Madison Avenue, New York, NY 10016, United States of America

British Library Cataloguing in Publication Data

Data available

Library of Congress Control Number: 2025933530

ISBN 9780192844835

DOI: 10.1093/oso/9780192844835.001.0001

Printed and bound by CPI Group (UK) Ltd, Croydon, CR0 4YY

The manufacturer's authorised representative in the EU for product safety is
Oxford University Press España S.A., Parque Empresarial San Fernando de Henares,
Avenida de Castilla, 2 – 28830 Madrid (www.oup.es/en or product.safety@oup.com).
OUP España S.A. also acts as importer into Spain of products made by the manufacturer.

Foreword

I greatly enjoyed reading Kotseruba and Tsotsos's very useful journal article (Kotseruba and Tsotsos, 2020) providing a review of cognitive and agent architectures. I also enjoyed and profited from reading this longer version, *The Computational Evolution of Cognitive Architectures*, which updates and extends their survey. This book provides a comprehensive review of the literature that includes many useful references and summaries of work along with a framework for interpreting what has been accomplished so far. Their expanded survey is the largest, most comprehensive, insightful, and informative to date.

This book will inspire further work. It provides much useful material for thinking about work in this area and seeing where progress has been made. Equally or perhaps even more importantly, it will help the reader to see where progress has not been made and where the reader can find areas to make progress based on the reader's interests, capabilities, and resources. In my case, it has suggested changes to two books and three projects I'm working on now (e.g. Ritter & Serdiuk, 2024). Thus, this book will be useful for modeling courses, for advanced modelers, and those wanting a survey of the field. The deeper you look into it, the more you will find.

Frank E. Ritter
frank.ritter@psu.edu
October 16, 2024

Contents

II HOW ARE COGNITIVE ARCHITECTURES BUILT?

Figures

Tables

What is this book about?

This book is a result of many years of surveying cognitive architectures and building a cognitive architecture called STAR that grew out of the successful Selective Tuning model of human visual attention proposed by John Tsotsos from the late 1980s onwards.

To inform our own research, we were interested in the past and current efforts towards building cognitive architectures. We found many excellent surveys, but they were either too narrowly focused or out of date. Even compiling a list of cognitive architectures proved to be a challenge as this information was spread across numerous disparate sources. Therefore, we conducted our own broad review of the field. The result of this work is our account of how cognitive architectures evolved in parallel with artificial intelligence (AI) and cognitive science, what goals were pursued by their creators, what paths have been taken, what has been achieved, and what lies ahead.

Contents of the book

To ensure objectivity and expand the scope of our review, we adopted a data-driven approach. We began by selecting a representative sample of cognitive architectures from hundreds of projects. Then, we gathered thousands of publications on these projects and carefully studied them to categorize architectures and their properties based on the existing and newly defined taxonomies. These annotations provide the quantitative foundation for our analysis, which includes diagrams and tables that offer a big picture of the topics covered in the book. However, not everything could be quantified in this manner. Therefore, in addition to visualizations and statistics, we summarized and analyzed theory and implementation of cognitive architectures in more detail. Here too, we attempted not only to discuss theories and algorithms but also provide the necessary background for their origins. To do so, we surveyed hundreds more reviews, books, research papers, and magazine articles covering a range of topics in philosophy, psychology, neuroscience, computer science, and robotics.

The field of cognitive architectures developed at the intersection of many disciplines. It combines and refines a multitude of perspectives on the nature of human cognitive abilities to produce complex software and hardware artifacts, hundreds of which have been developed to date. However, this book does not attempt to discuss each architecture in isolation. Instead, we group common themes aggregated across dozens of projects into four parts:

Part I is dedicated to the overview of cognitive architectures, what they are about, where they come from, and what kinds there are.
Part II goes into details of the individual components of cognitive architectures that correspond to core human cognitive abilities—perception, memory, learning, and reasoning. In the final chapter, we explore how these components are arranged and interact within the typical

Part III answers a simple question—what can cognitive architectures do? Here, we identify and categorize hundreds of applications of cognitive architectures. We also look at the methods for assessing performance of cognitive architectures in qualitative or quantitative terms.
Part IV examines recent developments in artificial intelligence and their implications for cognitive architectures. The book concludes with a summary of the challenges faced by the field of cognitive architectures, based on the views of the community and our own observations, as well as promising directions and best practices for future research.

Within each of these parts, individual chapters go into more specific details of the corresponding topics. While the content of each chapter differs, a consistent structure is maintained throughout, as follows:

- Each chapter begins with the summary of its contents and key sections within it, providing an overview of what is to come;
- The first section is usually dedicated to the definitions of basic terms and taxonomies that frame the discussion that follows;
- Next is a high-level overview of the topic, often accompanied by visualizations to add structure to concepts;
- The bulk of each chapter focuses on the detailed descriptions and analyses of the approaches to modeling particular human cognitive abilities. Typically, many specific examples of theoretical and algorithmic solutions found in different cognitive architectures are provided;
- Each chapter concludes with the summary of the most salient points discussed.

Who should read this book?

We hope that the information in this book will interest a wide range of readers who want to learn about the progress towards modeling human cognition. The book is particularly geared towards those who work with and develop their own cognitive architectures or are involved in the field in some capacity.

As the number of cognitive architectures continues to grow and existing ones evolve, keeping track of them all becomes increasingly difficult. Given that these architectures stem from a wide range of disciplines, including biology, computer science, and philosophy, the literature is dispersed across many publication outlets and is challenging to navigate.

This book aims to bridge this gap by providing a comprehensive guide and a reference for theoretical foundations and engineering aspects of cognitive architectures. Due to the breadth of the material covered, the use of specialized terms could not be avoided. Therefore, a certain level of familiarity with research in artificial intelligence prior to deep learning (e.g. topics covered in Russell and Norvig's (2020) textbook) is expected, as well as basic understanding of the terminology and methodology used in psychology, neuroscience, philosophy, and cognitive science.

Acknowledgments

This book would not have materialized without efforts of many people. First, we would like to thank Frank E. Ritter who championed this project. His guidance, encouragement, and advice throughout the publishing process and

extensive comments on the draft of the manuscript were invaluable. We appreciate the support of Martin Baum, Joan Bossert, and Karen Bunn at Oxford University Press. Many thanks go to our copy editor Joy Mellor for her meticulous and thorough work on the manuscript and to our project manager Priyanga Velmurugan at Integra Software Services. We are grateful to Amir Rasouli for thorough proofreading multiple drafts of the book, many helpful discussions, and lots of useful suggestions. Lastly, we appreciate the thoughtful and constructive feedback we received from Gary Bradshaw, Christopher Dancy, Hillmer Chona, Ion Juvina, William G. Kennedy, Robert St. Amant, Alexei Samsonovich, and Jun Tani. While this book benefited greatly from the expertise of reviewers, we take full responsibility for any mistakes in the text.

Part I

COGNITIVE ARCHITECTURES: AN OVERVIEW

We begin the book with three chapters introducing cognitive architectures. Chapter 1 discusses the definitions and goals pursued by creators of cognitive architectures and lists the projects that will be discussed in the rest of the book. Chapter 2 is dedicated to the history and development of cognitive architectures at the intersection of artificial intelligence and cognitive science. In Chapter 3 we categorize the set of selected cognitive architectures based on their type, internal representations, and cognitive conformity.

1 What Are Cognitive Architectures?

Cognitive architectures emerged in the early 1970s as a response to a dominant paradigm in computer and cognitive science that pursued modeling isolated aspects of human intelligence. Proponents of cognitive architectures sought a more holistic view of the human mind and its operation that integrated theoretical knowledge with experimental data and could be implemented as a software artifact. The premise of this approach was that a combination of theory and engineering could help identify gaps in knowledge, make new predictions, test their validity, and aid in developing cognitively plausible computational solutions.

In practice, however, designing and building cognitive architectures proved to be no easy feat, requiring multidisciplinary expertise in addition to formidable programming skills. But, perhaps, the most persistent challenge was posed by the inherent ambiguity of the core concepts, such as intelligence and cognition, as well as the notion of cognitive architecture itself. Therefore, in this introductory chapter we will begin by fleshing out the definitions, goals, and key properties of cognitive architectures.

Section 1.1 describes the original proposal of cognitive architectures and its intent.
Section 1.2 details what cognitive architectures model, at what level of abstraction they operate, and what properties they must possess to represent various aspects of human cognition.
Section 1.3 discusses the theory and implementation aspects of cognitive architectures.
Section 1.4 investigates the differences between cognitive architectures, their instances, and the underlying software infrastructure needed for their functioning.
Section 1.5 gives an estimate of how many cognitive architectures have been developed to date.
Section 1.6 defines inclusion criteria for the cognitive architectures that form the basis for this book.

1.1 Definition and motivation for cognitive architectures

Cognitive architectures are often called "blueprints" for human cognition (Duch et al., 2008; Bach, 2009; Vernon, 2014) because they define the building blocks that correspond to various human cognitive abilities and outline the information flow among them. Although the term "cognitive architecture" first appeared in print in (Anderson, 1983a, p. ix), the concept is attributed to Allen Newell. The General Problem Solver (GPS), which he developed together with Herbert Simon and John Clifford Shaw in 1959, is widely regarded as a precursor to cognitive architectures. Newell's

Cognition is a culmination of many years of research and remains one of the most important texts in the field.

1.1.1 Beyond binary questions

A better understanding of cognitive architectures can be gained from examining the rationale behind their development as described in a seminal paper by Newell (1973b). There, he outlined an alternative to the prevailing research paradigm in cognitive science, which focused on investigating cognitive phenomena in isolation. Newell compared it to the classic verbal guessing game of twenty questions, where one player thinks of an object and others try to identify it by asking yes-no questions (is it small? does it move?). In this analogy, scientists play against nature by asking questions and obtaining answers through empirical studies with two types of variables: independent and dependent. The experimenter varies the conditions specified by the independent variable and records the changes in the outcome (dependent variable). Next, statistical test is applied to determine whether these observed changes happened merely by chance or were actually caused by manipulating the conditions. Depending on the findings, the theory is revised, new questions are asked, and further experimentation follows.

Although this methodology generated large volumes of empirical data, Newell argued that a coherent theory of the human mind was unlikely to emerge from a collection of disjoint factoids. Time has validated his concerns, as there is still no unified theory of cognition today. In the meantime, lack of effective knowledge accumulation has been attributed to three key issues.

Cognitive phenomena are hard to define and quantify. Physics is arguably one of the greatest examples of successful integration of empirical data in science. Rakover (2020) argues that this was only possible because a correspondence exists between theoretical and observational terms. Specifically, physics operates with the International System of Units (SI) based on seven fundamental units of measurements (e.g. kilogram, meter). Therefore, physical concepts, like length, can be measured in standard units that in turn can be manipulated mathematically; among several sticks of varying lengths the longest one will have the highest numerical value assigned, their lengths can be added, etc.

In contrast, psychological concepts are more complex, multidimensional, and open to interpretation. While some have operational definitions, there are no standard units of measurement. One can assign numerical values to cognitive concepts, such as IQ score to measure intelligence. However, there is no guarantee that the relationships among the values will reflect the relationships among the concepts—a person with an IQ of 100 is not twice as smart as a person with an IQ of 50.

Experiments lack precise definitions. The established experimental paradigm appears to be ill-suited for studying cognitive phenomena. As Almaatouq et al. (2024) note, complexity of the human cognition calls for complex experiments with multiple dependent and independent variables. This makes the space of possible experiments very large. Worse still, it is difficult to place experiments precisely within this space because study descriptions often omit important details and assumptions. Consequently, it becomes nearly impossible to determine why results from different studies contradict one another

as many potential factors could contribute to the outcome: the experiment designs could be different, one of the results could be wrong, or the observed differences in results could point to a genuine distinction. Similarly, when multiple studies yield comparable findings, it may be unclear whether the similarities reflect shared experimental methodology or properties of cognitive phenomena. This uncertainty is further compounded by the imprecise definitions of the concepts themselves.

Experiment data can fit infinitely many theories. Lastly, even assuming that experimental results are valid, deriving a theory from data is not trivial. Roberts and Pashler (2000) and Garcia-Marques and Ferreira (2011) argue that for any set of observations, there may be infinitely many functions that can be fit to it and as many theories that can provide an explanation. At the same time, there is no certain way of deciding which function or theory is the best. Goodman's (1965, p. 309) "new riddle of induction" illustrates this point. Given the empirical data that natural emeralds are green, one can reach the obvious conclusion "all emeralds are green." But it is also possible to come up with an arbitrary statement "all emeralds are grue," where "grue" means "green until some time t and blue thereafter." Similarly, for any given experimental result, it is difficult to decide which of the possible hypotheses is more likely.

The solution to these problems proposed by Newell (1973b) was to guide the process of questioning the nature by constructing cognitive architectures that would provide a theory of the human mind to *integrate* multiple experiments, *unified* high-level "grand" theories with low-level experimental results, and could be implemented *computationally*, i.e. via software or hardware artifacts.

Each of these requirements contributes to mitigating the issues described above. For example, implementation naturally forces one to be more specific when defining a theory and eliminates approaches infeasible in practice. Integration aims at a more holistic view of cognition by combining observations about its different parts and inferring what influence they have on one another. In contrast, when only a sliver of the mind's activity is considered at a time, connections to other phenomena are often approximated or omitted. Lastly, the unification requirement posits that a single theory should explain a wide range of cognitive phenomena using a minimal set of mechanisms and parameters, i.e. it should be as simple and general as possible.

Thus, cognitive architectures with elegant structures and few intrinsic parameters that explain multiple observations are preferable to narrow models that are tailored to specific data. These top-down theoretical constraints can limit the explosion of possible explanations (Van Rooij, 2008). By design, cognitive architectures are also less likely to overfit to any specific dataset because they abstract minute variations in data and focus on general properties of multiple observed phenomena (Hélie and Sun, 2014).

In addition to the integration, unification, and computability requirements, Newell stipulated that such systems must be *fixed structures* and perform *symbol manipulation* (Newell, 1990, p. 80).

The requirement that cognitive architectures remain stable over time is largely inspired by the distinction between hardware and software in computers. Hardware in a given computer is fixed compared to the contents of the registers and software that change more frequently. Similarly, cognitive

architectures specify the fixed modules and relations among them as well as representations for encoding problems and generating behaviors. These aspects of the architecture evolve slowly (on the order of tens of seconds to years), via changes introduced by the developers or as the architecture itself accumulates knowledge during operation.

Cognitive architecture on its own cannot do much; only when provided with the task and knowledge base can it generate output. Thus, the contents of the modules is not predetermined and can often change during operation. However, momentary changes in memory contents do not impact the structure of the system.

The requirement of symbol manipulation or the "physical symbol system hypothesis," first formulated by Newell and Simon (1976), states that a symbolic system is necessary and sufficient for producing general intelligent action. The term "physical" refers to the physical realization of the computation (in a brain or a computer), symbols are patterns (or tokens) that can be used to form expressions, and "system" is viewed as an entity capable of running processes that manipulate symbols according to a predefined set of instructions. This hypothesis was not derived logically, but rather through generalizing from empirical evidence: the necessity of symbolism was attributed to the successes of symbolic artificial intelligence (AI) in a variety of domains, whereas cognitive science provided human data that supported hypotheses of symbolic processing and was useful for building symbolic models.

To sum up, cognitive architectures are theories of human cognition that are computationally implemented as relatively stable structures, can be applied across many tasks, and operate on symbols. Half a century later, these fundamental requirements still guide development of the cognitive architectures, except for symbolism, which has since been contested. As we will see by the end of Chapter 3, the term "cognitive architecture" now extends to a much wider range of systems than Newell and his colleagues had originally envisioned.

1.1.2 Bridging levels of abstraction

Integration, the core component of cognitive architectures, is a way of reconciling models of cognition different levels of abstraction. To separate conceptually different ways of describing the mind, Newell (1982) proposed the hierarchy with the *knowledge* level at the top, the *symbol (program)*, *register*, *logic*, and *circuit* levels in the middle, and the *device level* at the bottom.[1]

The top knowledge level is concerned with the function and purpose of the computation. At this level, the agent's goals and actions to achieve them are specified. The middle levels in Newell's proposal are modeled after computer system hierarchies and reflect his physical symbol system hypothesis. The bottom device level defines what hardware the computation is running on (e.g. a computer or a biological brain).

[1]The idea of separating theory from implementation is usually credited to David Marr. Much like Newell (1973b), who advocated for integration of theory and experiments in cognitive science, Marr aimed to lead computational vision away from merely designing data structures and algorithms to solve problems and toward more holistic understanding of computation. To do so, Marr proposed the following three-level hierarchy with *computational theory* (describing "what" and "why") at the top, *representations and algorithms* ("how" to solve the problem) in the middle, and *hardware implementation* (e.g. in a computer or a brain) at the

In this hierarchy, the levels in the middle are fully reducible and independent of each other. This enables program development at multiple levels in parallel. For example, programmers do not need to know how to design logic circuits to write code and managers can give tasks without knowing how to program. Thus, reducibility of levels ensures that a high-level description of the program can be expressed in terms of lower levels, e.g. assembly instructions, binary code, all the way to physical transistors.

However, the highest knowledge level cannot be reduced to the lowest device level entirely because some concepts simply do not exist there. Because more information is available at higher levels, reduction down the hierarchy is underdetermined, i.e. the mapping between theoretical concepts and implementation is not one-to-one but rather one-to-many. For instance, a general concept of sorting can be implemented by many distinct algorithms with different properties.

Levels of abstraction applied to cognitive architectures provide a framework that merges theory with empirical results. Theoretical descriptions of the fixed structures and processes that support intelligent behavior reside at the knowledge level. At this level, one can use human performance on behavioral tasks to explain the purpose of the computation and define desired inputs and outputs. The target at the bottom level is represented by neural models and data. Both the abstract ideas at the top level and physical constraints at the bottom level help select appropriate representations and algorithms in the middle level. By ensuring that algorithms and representations process information and produce outputs consistent with behavioral and neural data, cognitive architectures effectively bridge all three levels (Cooper and Peebles, 2015; Griffiths et al., 2015).

1.1.3 Reverse engineering the mind

Cognitive architectures can be viewed as attempts to reverse engineer the human mind by creating artifacts with human-like behavior and internal structure. The idea of reverse engineering ultimately rests on the assumption that cognitive processes that enable intelligence are tractable and can be specified and articulated as a computer program. At the core of this assumption is the recognition that the human brain has finite computational capacity, which puts constraints on what it can compute. One possible formalization of such constraints is given by Van Rooij (2008) as polynomial-time computable, which also includes super-polynomial computations confined to small input parameters (referred to as fixed-parameter tractable by Downey and Fellows, 1995). However, not all ascribe to this view. A different perspective is given in Landgrebe and Smith (2022), who gather evidence from mathematics, computer science, and biology to show that cognition (even at the level of vertebrates) is fundamentally out of reach for computational modeling on the grounds of its complexity.

Presuming that the mind is tractable and can be described in computational terms, one can start either at the top or the bottom of the hierarchy. The issues with the bottom-up approach of collecting and aggregating empirical evidence were discussed earlier. To that, we add an interesting experiment that illustrates the practical limitations of such empirical methods. Jonas and Kording (2017) tested the effectiveness of neuroscience methods by applying

them to a known man-made system—the MOS6502 microprocessor. Although some understanding of the microprocessor was gained through connectomics, lesions, analysis of tuning properties, local field potentials, and other common techniques, the information obtained in this manner still was not sufficient to fully reconstruct the original circuit. It is likely that similar efforts may fail or result in suboptimal designs when applied to evolved natural objects that are more complex and where many-to-many relationships exist between elements of the system and their functions.

The process advocated by Newell (1982) is to start with the specification of the agent at the knowledge level and working toward the implementation. This path is not without downsides either as it ultimately rests on the optimistic assumption that human cognitive function is understood and can be described in a way that is not too general, not too limiting, and computationally feasible.

In summary, for a top-down model of cognition, one needs to precisely specify the function to be computed, which may not be possible. Working from the bottom layers toward the top is also problematic. The immense amount of accumulated experimental data does not cover all phenomena and at times presents contradictory evidence, which is hard to analyze due to the inherent ambiguity of concepts and lack of reliable techniques for statistical analysis, leaving gaps in our knowledge.

Thus, cognitive architectures are typically constructed via an iterative process that combines both theorizing and experimentation (Anderson, 1991). The development of the architecture starts with the hypothesis, based on which a prototype is implemented. The next step is evaluation which may reveal errors and biases in the implementation. These issues are then corrected by modifying the implementation, updating the hypothesis, or bringing in new data. Ideally, multiple hypothesis-implementation-evaluation-update cycles would simultaneously expand architecture's explanatory capabilities and unify the underlying empirical observations into a coherent whole. In reality, as the rest of this book will demonstrate, every step described above is rife with non-trivial problems and solutions that lead to dead ends.

1.2 What do cognitive architectures model?

Nearly all cognitive architectures declare that modeling aspects of human cognition is their goal, but few specify precisely what it means. Often the definition of cognition as human-level intelligence is implicit in references to other artificial and biological systems, as well as high-level descriptions of the characteristics and abilities of the architectures.

In this section, we will look at psychological and computer science definitions of intelligence that inspired most cognitive architectures, discuss at what level of abstraction intelligence is modeled, and what properties are considered essential for intelligent behavior.

1.2.1 Toward human-level intelligence and beyond

The concept of "intelligence" is widely used and easily understood; yet, a formal, precise, and general definition of intelligence has remained elusive. Because what is colloquially considered intelligent is rather broad, there are a

growing number of proposals that emphasize different aspects of intelligence. Historically, intelligence has been investigated across many disciplines, often independently of one another, resulting in many context-specific proposals and definitions (see the comprehensive list in Legg and Hutter, 2007a).

Intelligence in psychology

In psychology, research on intelligence has mainly focused on identifying distinct cognitive abilities and measuring them through carefully designed tasks and assessments. The most established theory of human cognitive abilities with the largest amount of empirical support to date is the Cattell-Horn-Carroll (CHC) theory (Flanagan and Dixon, 2014). It is a combination of the fluid-crystallized theory developed by Cattell (1963) and Horn (1991) with Carroll's (1993) three-stratum theory of cognitive abilities.

The original Cattell's theory listed two primary factors: fluid intelligence (inductive and deductive reasoning, and learning determined by biological factors) and crystallized intelligence (acquired knowledge abilities determined by social and cultural factors). This model was further expanded by Horn to include eight abilities: visual perception, auditory perception, short-term memory, long-term memory, speed of processing, reaction time, quantitative, and reading-writing.

Carroll's (1993) three-stratum theory is based on an extensive collection of quantitative experimental data related to various aspects of human performance across hundreds of abilities and tasks. Through factor analysis applied to this data, Carroll identified hierarchical relationships between general intelligence, abilities, and tasks.

Stratum I, the lowest in the hierarchy, contains a large number of narrow abilities (e.g. induction, reading comprehension, perceptual speed) that can be directly measured through psychological tests. These abilities are subsumed by broader Stratum II abilities, including fluid and crystallized intelligence, perceptual abilities, memory retrieval, and cognitive speediness. Finally, at the apex of the hierarchy, Stratum III represents general intelligence (or simply the g factor).

Strata locations within the hierarchy are related to the order of factor analysis. First-order factors result from analysis of correlations between the original values in the experimental results. Second-order factors emerge from examining correlations between first-order factors, and so on (Carroll, 2005).

Despite differences between the Cattell-Horn and Carrol's taxonomies, for practical reasons they were later integrated by McGrew (2005) into a single CHC theory[2] (Wilhelm and Kyllonen, 2021). For nearly three decades CHC served as a dominant framework for studying human intelligence. It has primarily been used to design, administer, and interpret numerous intelligence tests for education, theory development, and clinical diagnoses (McGrew, 2023). As such, it is currently the most complete and continuously updated taxonomy of human cognitive abilities (Flanagan and Dixon, 2014; Wasserman, 2019).

Intelligence in computer science

Definitions of intelligence in computer science and AI were influenced by advancements in cognitive science but even more so by a number of practical

[2]See a visual tour of CHC by Schneider and McGrew

considerations, such as utility, adaptability, robustness, and response time of the artificial systems. These properties are a common thread in many definitions of intelligence from the AI standpoint (Winston, 1992, p. 5; Legg and Hutter, 2007a; Luger, 2009, p. 8; Russell and Norvig, 2020, p. 36).

Given the computational nature of most AI efforts, a formal mathematical definition of general machine intelligence is highly desirable for more principled development and evaluation of intelligent systems. So far, the most consistent push in this direction was made by Legg and Hutter (2007b). From a large sample of over seventy definitions of intelligence found in encyclopedias, computer science, and psychological literature, Legg and Hutter (2007a) derived a verbal definition of intelligence as *the property of an agent that can sense and interact with the environment, succeed in pursuing goals, and adapt to changing objectives and conditions.* They also proposed a mathematical formalism based on this definition, which represents the universal intelligence of an agent π in terms of environment μ, goal (implicit in the environment), and the agent's ability to achieve the goal, as given by the value function V_μ^π (Legg and Hutter, 2007b):

$$\Upsilon(\pi) := \sum_{\mu \in E} 2^{-K(\mu)} V_\mu^\pi.$$

However, an agent's performance depends not only on its abilities but also on the properties of the environment. In highly limited settings, even a very simple agent can trivially succeed. To account for this, an additional term $2^{-K(\mu)}$ is introduced in the equation above, which weights the agent's performance by the complexity of the environment.[3]

Despite claimed advantages, such as a clear mathematical formula applicable to a broad range of biological and artificial entities, this definition is not without issues. Perhaps, the most significant is the decision to fold the goal into the environment rather than attributing goal-setting to the agent. Furthermore, there are problems with applying this definition in practice because values of the numeric terms in the equation cannot be derived analytically. Instead, they must be estimated based on the sample of the agents working toward various tasks in a range of environments. Interestingly, this aspect is reminiscent of the approach taken in psychology, which likewise operates with estimates of human performance measured through experimentation.

Intelligence in cognitive architectures

Most cognitive architectures are created as models of human intelligence. However, for the majority of projects no concrete definition of human intelligence is provided. Below we provide examples of how intelligence was operationalized in some cognitive architectures by combining aspects of psychological and computer science definitions.

[3]Legg and Hutter in their definition incorporate the Kolmogorov complexity measure established in the information theory. Kolmogorov complexity K of an algorithm is defined as the length of the shortest computer program that reproduces it. As such, K can be applied to the description of the environment μ represented as a binary string. Then, $K(\mu)$ becomes the length of the shortest program that can produce the description of μ as output. Naturally, longer programs would be needed to describe more complex environments, reflected in a higher value of Kolmogorov complexity. When this is the case, the discounting factor in the authors'

Soar. General intelligence is defined as the ability to 1) perform tasks of varying complexity (from routine to open-ended), 2) use appropriate problem-solving methods and representations, and 3) learn about all aspects of tasks and its own performance on them (Laird et al., 1987). An extensive list of sample tasks is provided, for many of which human performance data is available for comparison.[4]

RCS. Intelligence is framed as the ability of a system to achieve goals in an uncertain environment by choosing actions that increase the probability of success (Albus, 1991). The elements of intelligence include sensors, sensory processing, world model, values, behavior generation, and actuators. Intelligence is measured by a vector of values that include 1) the computational power of the computer, 2) the complexity of the system's elements, 3) the information in the system's memory, and 4) the elegance of the system architecture. Notably, learning is not listed as necessary for intelligence, only to become more intelligent.

NARS. A working definition of intelligence is the ability for an information-processing system to adapt to its environment with insufficient knowledge and resources (Wang, 1995b). Here, an information-processing system consists of formal language, semantics, inference rules, memory, and a control mechanism. Insufficient knowledge and resources place the following restrictions: finite information-processing capacity, real-time response, and an open task and knowledge (any that can be represented in the system's language).

CogPrime. General intelligence is defined as the ability to achieve complex goals in complex environments, using limited resources (Goertzel et al., 2013). The mathematical formalism for this definition is derived from the Legg and Hutter's (2007b) equation that we saw earlier, with additional terms to express goals and computational resources (Goertzel, 2010). Like Soar, CogPrime has an associated list of tasks that represent the space of intelligent behaviors.

While it is not always stated, human-level cognitive abilities may still be implicitly expected of cognitive architectures, given that any system capable of modeling and explaining human cognition should possess similar qualities. In this context, the definition of intelligence becomes part of the solution—identifying sufficient and necessary properties of intelligence is key to understanding its nature, modeling its workings, and evaluating the outcomes.

One important factor to consider is what reference point to set for human level of intelligence. By definition, most humans possess human-level intelligence, yet their individual abilities vary significantly, resulting in a wide range of performances when treated as a population. Therefore, when attempting to quantify human-level intelligence, three general approaches can be taken: averaging the performance of multiple individuals, determining a range of performance that applies to the majority of the population, or considering only the top performance achieved within the population. Psychological research primarily concentrates on the first two—establishing an average performance

[4] ACT-R, EPIC, ICARUS, and PRODIGY are broadly considered as belonging to the Newell tradition and thus likely use a similar definition of intelligence. DAC, although it is not commonly listed as a part of the group, recasts Soar's definition of intelligence within a Bayesian decision-making framework (Verschure and Althaus, 2003).

or performance range for various groups of humans on specific tasks. In contrast, the field of AI focuses on matching or surpassing top human performers across multiple tasks and abilities.

Goals and conceptual frameworks of many cognitive architectures are a blend of approaches borrowed from psychology, neuroscience, and computer science. For instance, intelligent behavior is often framed in the AI tradition as action, inference, and learning for achieving goals. However, when it comes to evaluation, the emphasis is skewed toward matching a typical human behavior rather than surpassing the top-performing humans, especially in architectures influenced by psychology and cognitive science. Chapter 10 revisits the topic of evaluating cognitive architectures in more detail.

1.2.2 Desiderata

Besides definitions of intelligence and other core concepts, many creators of influential architectures also formulated lists of desirable features or desiderata for short. The first such list was proposed by Newell (1980) and another dozen or so lists by other authors appeared during the next several decades. These desiderata often represent personal preferences embedded in respective projects and may not necessarily apply to the entire field. But despite their somewhat subjective nature, these lists reveal what aspects of the cognitive architecture development were considered important by their designers.

Due to the differences in how each author expressed their views, there is no common format for desiderata. As a result, the same concepts appear in different lists at different levels of generality (e.g. learning vs. reinforcement learning) and from different points of view (e.g. cognitive vs. biological). Certain items, such as real-time operation, autonomy, or self-awareness, are not precisely defined and operationalized, while definitions of others may overlap (e.g. development, adaptation, and learning refer to aspects of the same process).

With these considerations in mind, we grouped similar desiderata from twelve lists into broader topics and summarized how often each theme was addressed across lists Table 1.1. We will now briefly discuss each group, starting from the most frequently mentioned.

Learning. The ability to learn is considered by many as essential for intelligent systems, therefore this property is included in more than half of the lists. In particular, Newell (1980; 1989) emphasized the importance of learning from the environment, experience, and via developmental processes. Others focused on specific learning mechanisms, e.g. bottom-up (Sun, 2004), perceptual (Vernon et al., 2016), error-driven (O'Reilly, 1998; Sun, 2004), Hebbian (O'Reilly, 1998), and meta-learning (Thórisson and Helgasson, 2012). It is often acknowledged that a combination of multiple learning mechanisms would further improve the adaptability and robustness of the system.

Hardware and software design. Implementation is a substantial part of any cognitive architecture, and there are a number of things to consider when designing hardware and software. For example, Krichmar (2012) advocates for biologically motivated principles initially proposed by Pfeifer and Bongard (2006) for cognitive robotics. According to these principles, robot design should be appropriate for its niche and tasks to ensure the balance between the

Table 1.1 A table listing desiderata grouped by topic (rows) and lists of desiderata (columns). Plus symbols ('+') indicate that one or more desiderata in the list belong to the corresponding topic. Topics are sorted by the total count of corresponding desiderata across all lists.

	Newell1980	Newell1989	Wooldridge1995b	Pack1997	Medeiros1998	OReilly1998	Sun2004b	Laird2010	Krichmar2012	Thorisson2012	Varma2014	Vernon2016
Learning	+++	++				++	++	++		++		++
Software and hardware design				+	+++		++++	+	+++++			
Representation	++	++				+++	+	++++			+	
Knowledge & reasoning	+	+		+	+		++	+++				++
Goal-driven behavior	+	+	++	+	++							++
Biological realism	++					+	+				+	
Reactivity	+	+	+		+							
Value system			++						+			+
Resource management				+	+					+		+
Motor control		+	+					+				
Social skills		+	+	+								
Real-time operation	+	+								+		
Robustness & reliability	+				++							
Meta-cognition	+	+						+				
Autonomy	+	+	+									+
Perception	+	+			+							
Embodiment									+			+
Sensory-motor coordination									+			+

complexity of the agent's perception, motor and neural systems, and the properties of the environment. Although cheap design is a basic principle in biological systems, it is rarely exploited in the existing cognitive robots that require extensive computation, consume too much power, and often fail to take into account the properties of their domain.

With regard to software, Medeiros (1998) prioritizes ease of development and suggests modular, flexible, and expandable design. Modularity has additional advantages, e.g. it allows encapsulating mechanisms functionally and anatomically (Sun, 2004) and accommodates diverse implementation methods for different subsystems (Pack et al., 1997; Sun, 2004). Krichmar (2012) argues for more biologically motivated modularity, where parts of the system are loosely coupled and perform overlapping functions to increase the overall redundancy.

Representation. The choice of representation has numerous consequences for the design of the entire system. Therefore, it is not surprising that multiple desiderata relate to the structure and function of internal representations. What representation is appropriate for modeling cognition remains a highly contended topic. Those who worked on symbolic architectures promoted rule-based approaches that supported the use of symbols, abstractions, and natural language (Newell, 1980; Laird and Wray III, 2010), whereas others argued for the use of subsymbolic representations, such as neural networks, that were deemed more flexible and biologically realistic (O'Reilly, 1998) or hybrid representations combining both symbolic and subsymbolic elements (Sun, 2004). The advantages and disadvantages of these representations are covered in more depth in Section 3.3.

Knowledge. Reasoning and decision-making rely on information that the intelligent system has at its disposal. A large, extendable, and easily accessible

knowledge base is a must for most non-trivial applications (Newell, 1980; Newell et al., 1989; Laird and Wray III, 2010). Typically, knowledge is stored in and managed by a dedicated memory system, often subdivided into more specialized subsystems that store factual information, personal experiences, procedural skills, and sequences of motor actions (see Chapter 5).

Goal-driven behavior. Setting and completing goals is considered by many as the basis of intelligent behavior. Wooldridge and Jennings (1995) refer to it as proactive, i.e. taking an initiative to satisfy some objective, as opposed to reactive behavior that responds to the changes in the environment. Medeiros (1998) similarly notes that reactions to external changes should be guided by the objectives of the main task. An intelligent system should deliberately choose its goals and commit to them instead of pursuing a single hard-coded task. According to Pack et al. (1997), these goals must evolve over time. Newell (1980) and Vernon et al. (2016) point out the prospective nature of goal-setting, such that the system anticipates the desirable outcome and strives to achieve it. Vernon et al. (2016) also emphasize the importance of goal-setting in the developmental context, citing psychological studies that observed infants performing goal-directed actions to develop sensorimotor abilities.

Biological realism. Although most designers of cognitive architectures strive to achieve some degree of biological realism, there is no strict meaning of what it entails. Newell (1980) in his set of desiderata stated that the system should be realizable within the brain, but noted that our understanding of what the brain is and does is subject to change. Varma (2014b) suggests that biologically realizable systems should be merely consistent with the neural information processing instead of directly mimicking it. According to O'Reilly (1998), biologically realistic systems should provide insight into how brain processes give rise to cognition. Thus, any cognitive processes should be informed and constrained by the evidence provided by neuroscience, and any mechanisms that are not consistent with what is known about the brain should be avoided even if they perform a similar function. Complexity constraints apply here too—any computation that exceeds resources available in the brain must be modified accordingly (Tsotsos, 2017).

Another aspect of biological realism is the evolutionary history of cognitive abilities. Sun (2004) suggests that a cognitive model of human intelligence should be reducible to models of animal intelligence. In other words, data structures and computational mechanisms proposed for human and animal cognition must not be drastically different but rather form a continuum.

Reactivity. While many desiderata focus on goal-setting as a signature of intelligent behavior, reactivity is equally necessary for adaptation and timely response to changing environment (Wooldridge and Jennings, 1995; Medeiros, 1998). Newell (1980) argues for reactivity as a way of adjusting to the demands of the environment, which cannot be achieved if the system is purely goal-driven and unresponsive to the external influences.

Value system. The value system should at least include the basic understanding of what is good and bad for the artificial entity itself. This property is present in virtually all biological organisms and according to Krichmar (2012), is key to learning and acting within the environment. However, implementing intrinsic values is difficult since they are dissociated from the artificial body

(real or simulated), whereas in biological systems, pain, pleasure, hunger, and fatigue drive a true value system. Thus, a biologically realistic implementation would require the incorporation of the neuromodulatory system into decision-making. Vernon et al. (2016) stress the importance of basic drives in shaping the intelligent system through development. Wooldridge and Jennings (1995) further suggest ethical aspects, such as veracity and benevolence. The former guarantees that the intelligent agent will not knowingly lie, and the latter that the agent will try to do what is asked of it.

Resource management and attention. Intelligent systems placed in relatively complex environments have to deal with the constant stream of incoming information, multiple goals, and limited resources. A number of resource management mechanisms have been suggested to tackle these problems. Attention is perhaps the most common one, which allows focusing on the events and objects that are salient or relevant to the current goals (Pack et al., 1997; Vernon et al., 2016; Thórisson and Helgasson, 2012). In the case of conflicting goals, the system must correctly prioritize them and select the most appropriate one for execution (Medeiros, 1998).

Motor control. The ability to control a physical or simulated body is needed for many tasks. Basic moving around is the minimum requirement (Wooldridge and Jennings, 1995). However, in order to perform multiple tasks of different types and varying complexity, a body with many degrees of freedom is necessary (Newell et al., 1989). Together, the multitude of tasks and the complexity of the body require rich control knowledge. One of the possibilities of organizing such knowledge is via a hierarchy where complex tasks are composed of primitive operations. Such organization is more efficient because it allows reuse of knowledge and consistent with the developmental paradigm, which proceeds from simple to complex tasks (Laird and Wray III, 2010).

Social skills. Although the ability to interact with other artificial agents or humans is arguably important, it is included only in three lists, namely, by Newell et al. (1989), Wooldridge and Jennings (1995), and Pack et al. (1997). The first one particularly emphasizes the social aspects of autonomy, whereas the other two consider the social skills on their own. There is little discussion on what social skills may be, only Wooldridge and Jennings (1995) speculate that they involve communication via language.

Real-time operation. Real-time operation is typically defined with respect to the passage of time in the external environment and the speed of events in it. As long as the system operates fast enough to respond to those events, it may be considered real time. Newell (1980; 1989) mentioned the real-time requirement as one of the constraints that affects all processes in the system. For example, it can put limits on the rate of input that the system receives and speed of its reaction. Consequently, a combination of incomplete sensory information and insufficient time for processing it can result in potentially suboptimal decision-making and prompt corrective actions.

Thórisson and Helgasson (2012) consider real-time operation as crucial for any autonomous embodied system that must act in synchrony with its environment. To them, four aspects of real-time operation are important: granularity (the amount of uninterrupted processing between accepting new input), reactiveness (efficiency of the sense-act pathway), temporal planning horizon

(for long-term tasks), and uptime of the system (continuous or intermittent operation).

Robustness and reliability. Besides being able to operate in normal conditions, the system must gracefully handle imperfect or incomplete inputs, unexpected events, and malfunctions (Newell, 1980; Medeiros, 1998). This is necessary for operating without catastrophic failures or performance degradation over the lifetime of the agent.

Self-awareness and meta-cognition. In addition to sensing and reasoning about its surroundings, an intelligent system must have some understanding about its knowledge, abilities, and limitations. In their desiderata, Newell et al. (1989) include self-awareness, roughly defined as the ability to model, reason about, and modify itself, although do not mention any direct implications for the design of the architectures. This is addressed by Laird and Wray III (2010), who point out the importance of meta-cognition for resource management, error correction, and learning. However, the necessary extent of introspection remains an open problem. For instance, humans do not have a complete knowledge about the state of their body and mind and often realize their deficits only after experiencing a new situation or performing a new task. Psychological evidence suggests that self-management and self-reflection skills of humans are not part of the established system, but rather a set of strategies contingent on individual experience and cultural background (Fletcher and Carruthers, 2012).

Autonomy. Autonomy, like cognition and intelligence, is a difficult concept to define. The literal meaning of the word is "self-governing," and the concept of autonomy as such is implied in some desiderata (e.g. goal-driven behavior, learning, meta-cognition, and value systems). Explicitly, autonomy is stated only in three lists, four if counting Thórisson and Helgasson's (2012) paper, where all desiderata are viewed as aspects of autonomy. Wooldridge (1999) explains autonomy as the ability of the system to operate without interventions from others and to control its own actions and internal state. Newell et al. (1989) consider two aspects of autonomy: independence of the environment and socialization. Although seemingly contradictory, interaction with other agents in the environment is a necessary requirement for developing and maintaining autonomy. Vernon et al. (2016) follow a similar definition but expand it further by separating behavioral and constitutive autonomy. The former defines the extent to which the system is independent of external influences, and the latter focuses on the ability to maintain itself through self-introspection and self-control.

Embodiment. Embodiment goes beyond the physical body and motor control discussed earlier. In other words, not every system that can operate on hardware is necessarily embodied. The primary distinction of embodied systems is that their physical structure and dynamics are an inseparable part of their cognition (Vernon et al., 2016). This implies that the body itself is embedded in the surrounding environment and should have adequate complexity and functionality for that environment (Krichmar, 2012). Thus, in the embodied entity, the brain, body, and the environment are tightly linked. Consequently, changes to any of these component affect the entire system: modifications to the body affect cognitive processes constrained by what the body can do, and

changes to the environment result in the appropriate adaptations in both the body and cognition.

Sensory-motor coordination (SMC). SMC refers to how biological organisms develop behaviors and acquire skills through interactions with the environment. The range of behaviors and skills obtained in this way can be quite varied, from simple gripping to mastering a musical instrument or driving a vehicle, or distinguishing between different objects. SMC is typically found in the embodied and developmental architectures (Krichmar, 2012; Vernon et al., 2016).

The coupling of perception and action is a drastic departure from the typical processing pipeline implemented in most artificial systems. As a result, SMC provides 1) physical control over objects, 2) naturally matches the level of perceptual and motor competence, 3) induces correlations in the sensory-motor space enabling figure-ground separation, object categorization, learning of affordances, and natural attention abilities, 4) integrates multiple sensory modalities, e.g. visual, haptic, proprioceptive, and 5) learns sensory-motor coordination itself from the signals produced by self-motion (Pfeifer and Scheier, 1997).

Perception. Despite the apparent need for perceptual input for any intelligent system, only two lists of desiderata mention this requirement explicitly. Newell et al. (1989) pointed out that architectures at the time did not elaborate perception sufficiently and instead worked with the preprocessed input. Although some progress toward more realistic perception has been made since, this problem is still far from being solved. While purely software systems can be provided with the input in any form, systems implemented in hardware must deal with physical sensors and the accompanying noise. As a result, most integrate input data from multiple complementary sensors to compensate for their limited accuracy and reliability (Medeiros, 1998).

The lists of desiderata discussed above reflect personal preferences in the development of cognitive architectures voiced by influential figures in the field. Taken together, they reveal areas of consensus that guided prior research and points of contention that can inform future efforts.

Overall, core properties of the intelligent system (general design and representations), cognition (acquisition and use of knowledge), and goal-driven behavior (presence of internal motivation) have been the main focus of the field since its inception. Other desiderata, such as biological realism, perception, embodiment, meta-cognitive abilities, systems engineering, have received less support historically but are gradually becoming more prominent. Furthermore, several important items, such as ease of system integration, reproducibility, and evaluation, are notably absent from all lists. This reflects the current state of the field, as will be evident from discussions in Part II and conclusions in Chapter 12.

1.3 From theory to software

Earlier, we described cognitive architectures as bridging multiple levels of abstraction, from high-level purpose of computation to the underlying physical

substrate. A similar hierarchy can be established for different stages of developing the architecture. The following categorization by Cooper and Guest (2014) is one such example that distinguishes between theories and implementations:

Conceptual theory. A theory is usually a high-level verbal description of the cognitive phenomena and processes modeled within the architecture (see Section 3.2).
Specification. Specification provides a more technical and precise formalization of theory, usually accompanied by information such as flow diagrams, characterization of inputs and outputs, and computational procedures for deriving outputs from inputs.
Implementation. The implementation of the theory is a source code, a binary file, or a physical device designed according to the specification using particular software/hardware tools.

Transitioning from high-level theories to practical implementations can be challenging, as verbal theories often focus on justifying their approach through references to other theories and concepts, while omitting technical details. Specification fills this gap by providing additional assumptions that may not be relevant to the theory but are essential for implementation. Such a specification should also permit validation and replication regardless of the availability of implementation. Finally, the implementation may contain additional assumptions that are specific to the programming language and libraries used to build it, but need not be included in the specification.

In general, as with the levels of abstraction, the mapping between theories and implementations is not one-to-one. For example, the same theoretical proposal may lead to vastly different implementations; a well-known Belief-Desire-Intention (BDI) framework proposed by Rao and Georgeff (1995) has served as a basis for numerous distinct agent, cognitive, and robot architectures, as has Baars's (1988) Global Workspace Theory. Similarly, a single data structure or piece of code can represent multiple concepts—graph structures are equally suited for symbolic knowledge representations (e.g. semantic nets) as well as distributed ones (e.g. models of neurons).

An additional source of variability is modifications to the architecture as it evolves over time, even if the underlying theory remains the same. To illustrate this, we will consider the MIDAS architecture implementing attention model developed by Wickens et al. (2003). In the original version, perceptual data and world knowledge were represented using a graph structure with nodes and links corresponding to concepts and relationships between them (Corker et al., 1997). However, in later versions, frames replaced semantic networks (Tyler et al., 1998).

The complex relationship between concepts, specification, and implementation requires additional validation steps to determine whether the specification truly represents the conceptual theory and whether implementation is correct with respect to specification. Validation should be ideally performed after any changes are introduced at any level of description.

In practice, given the complexity of the cognitive architectures, both descriptions and evaluation are lacking. Most cognitive architectures we reviewed are presented on the level between conceptual theory and specification. This lack of clarity poses a serious reproducibility issue, as the available information is often not sufficient to fully understand the theory and its formalization

needed for recreating the implementation, which is not available for many projects. In addition, evaluation procedures do not always follow best practices. We discuss this further in Chapters 10 and 12.

1.4 Distinguishing frameworks, architectures, and instances

Translating cognitive architectures into software or hardware artifacts requires extensive infrastructure. Besides implementing the components of the architecture, additional software is often needed for accessing and modifying the internal settings, visualizing intermediate processing results and final outputs, interfacing the architecture with external sensors or actuators, adding extensions to the architecture, evaluating its performance, debugging any software and hardware issues that may arise, and more. In most cases, implementation of these additional components has no relation to theoretical and structural commitments of the given architecture. But it does add another level of abstraction between the theory and its implementation. Below we will expand on the implementation level described in the previous section by discussing frameworks, architectures, and their instances.

Software framework. Frameworks are software tools for implementing abstract architectures.[5] These frameworks provide a high-level interface for specifying the design of components, connections among them, visualization, simulation environment or interfaces for hardware, support for additional libraries, various utility functions, etc., which greatly simplifies development and prototyping. Usually, few if any constraints are imposed on the internal organization of components or the type of computation within them.

Perhaps, the most known and mature generic framework is Robot Operating System (ROS) (Quigley et al., 2009). Although it can be used to implement an entire cognitive architecture, in practice ROS usually serves as a convenient interface between robotic platforms and existing cognitive architectures. For example, DIARC and LIDA leverage ROS to connect to simulated environments, use available robot models, and perform low-level operations, such as motion planning (Madl et al., 2016; Wilson et al., 2016). DIARC itself is implemented within another framework called Architecture Development Environment (ADE) (Scheutz et al., 2007). Another robotic framework RoboComp (Manso et al., 2010), tailored toward developing intelligent agents, has been used to build RoboCog and its descendant CORTEX, as well as a recent project MERLIN (González-Santamarta et al., 2020). The OpenCog framework (Hart and Goertzel, 2008) likewise aims to support an architecture-neutral development of integrated AI agents and is currently being used to implement OpenCogPrime.

Such frameworks are a relatively recent development compared to the timescale of the architectures we are considering, therefore the majority of the projects we reviewed, especially the early ones, were custom built from scratch. Projects developed in the past two decades tend to rely more on

[5]Many frameworks are themselves based on middleware, which typically sits between an operating system and applications to enable communication in a distributed system. Technically, middleware, forms another layer of abstraction, but we omit it here for

the existing software for common computations, such as navigation, basic perceptual operations, common algorithms, etc.

Architecture. Implementation of the core modules based on the verbal theories and specifications within a given framework and in a given programming language.

Instance/model/agent. In order to apply a cognitive architecture to specific tasks, additional information is needed, such as task description, input/output specification, knowledge base, parameters, interface with a robotic platform, etc. Therefore, a single cognitive architecture can give rise to multiple instances, which are also referred to as models or agents in the literature.

The range of abilities that a single instance of the architecture can demonstrate varies significantly. The most common models are developed to replicate specific aspects of human behavior, such as reaction time or eye movements, on narrowly defined tasks. Hundreds of such models have been built to simulate psychological experiments using ACT-R, EPIC, and CHREST (see Chapter 9 for more examples).

Models that perform more broadly defined tasks also exist. For example, Guardian for ICU patient monitoring (Hayes-Roth, 1996), Patient Advocate for in-home patient support (Miksch et al., 1997), and AIbots for office surveillance (Hayes-Roth et al., 1993) are all instances of the same AIS architecture. Similarly, the Subsumption architecture has been implemented on a number of custom built robots, such as Herbert, Tom and Jerry, Genghis, Seymour, and Toto, each equipped with different sensors and capable of performing different actions (Flynn and Brooks, 1989).

The objective of this book is to examine cognitive architectures not merely from a theoretical perspective but also through their practical implementations in specific instances. However, in the literature, theory, framework, and instances are often not clearly distinguished. This becomes a problem when the architecture has multiple implementations in different programming languages and multiple instances of each for different tasks. In this case, referring to models or instances using the same name as the architecture can lead to potential confusion. This brings us to the next topic—determining the number of cognitive architectures.

1.5 How many cognitive architectures are there?

There are very few estimates of the size of the field of cognitive architectures. Existing catalogs of architectures list up to several dozen projects. For our recent survey of the past forty years of cognitive architectures (Kotseruba and Tsotsos, 2020), we aggregated many surveys and lists, in addition to combing through thousands of results returned by academic search engines such as Google Scholar and the now defunct Microsoft Academic. This effort netted over 300 projects that were identified as cognitive, agent, or robotic architectures by their respective authors. However, this figure is also likely inaccurate because of the following factors contributing to the inconsistent naming and versioning of the cognitive architectures.

Unnamed projects. When a project does not have a consistent title, identifying related publications becomes very difficult. There are several architectures that are known by their author's names, e.g. the Haikonen cognitive

architecture (HCA) or Maes's behavior networks. However, in general, relying on the authors' names alone is not a viable option, as this approach is susceptible to changes in the research team.

Naming instances. Even if a project has an assigned title, there is no established convention for separating the architecture itself from its extensions and instances. As a result, they may be treated inconsistently, sometimes as separate entities and sometimes as a part of the architecture that spawned them. For example, Dav and Sail are two robotic systems based on the SASE cognitive architecture, but all three are often treated as different projects in the literature.

Versioning. Long-lived projects inevitably go through many changes and at some point begin to diverge from their initial form, in terms of both theory and implementation. There are three possibilities when dealing with such changes:
1) No change tracking. The majority of projects we reviewed do not track changes in the architecture itself or its implementation explicitly. Many do not have project pages or source code publicly available, some are too short-lived to require change tracking, and for others, the details are buried in publications and difficult to extract.
2) Name change tracking. Some projects assign a new name to major extensions, such as Metacat for a meta-cognitive extension of Copycat, LIDA for a learning successor of IDA (Intelligent Distribution Agent), and 4D/RCS for Dickmanns's (1990) 4D approach to vision integrated with the RCS (Real-time Control Systems) architecture.
3) Software versioning. Several projects explicitly maintain versions following software engineering conventions of assigning a number to major and minor releases, as well as patch updates.

Soar is one of the best maintained and documented architectures among the ones we reviewed. Each software release on the Soar project webpage[6] has detailed notes on the changes to the API and the architecture itself. In addition, a review paper by Soar's principal investigators Laird and Rosenbloom (2014) describes a history of Soar versions 1 through 6. According to these sources, some versions involved substantial changes in theory (e.g. introduction of subgoaling in Soar 2 and adoption of the single state principle in Soar 5), while others were mostly implementational (e.g. Soar 6 was a rewrite of Soar 5 for portability and efficiency).

Much less information is available for other architectures. For example, release notes on ACT-R and NARS are spotty, but reviews of their development by Ritter et al. (2019) and (Wang, 2006b, pp.354–356), respectively, fill some gaps. Other projects, such as BECCA, CLARION, CHREST, OMAR, MIDAS, and RCS, also use the software naming convention, but not all changes are documented consistently or described in published papers. Similar to Soar, the major versions of the architectures can entail either theory or implementation changes, or both.

In some cases, both names and versions of the architectures changed over time. For example, ACT-R (Adaptive Control of Thought—Rational), originated as the Human Associative Memory (HAM) model, which was later renamed to ACTE and ACT*, and, eventually became known as ACT-R, with

6

numeric versions starting with ACT-R 2.0 (Anderson and Lebiere, 1998; Ritter et al., 2019). Similarly, CLARION was initially based on the CONSYDERR architecture (Sun and Peterson, 1998a), with explicit versions starting with CLARION v4.[7]

External contributions. Complicating the matters further, sometimes projects can gain a substantial community besides the main research group. Contributions from external sources may extend existing theory, implementation, or application domain. For architectures like ACT-R, Soar, and ART, the additions are quite significant. However, it remains uncertain whether these extensions should be considered as part of the architecture, even if they align with the objectives of architecture's creators.

Broad definition. Besides the naming and versioning issues, the concept of a cognitive architecture has a very broad definition. As we will see later in Chapter 3, even projects such as ACT-R, Soar, CLARION, and EPIC, which are indisputable examples of cognitive architectures, have been referred to as different types of intelligent systems.

In sum, determining an accurate count of cognitive architectures is very challenging for the following reasons: a) unnamed projects are virtually impossible to trace in the literature, so their number remains unknown; b) inconsistent naming and versioning criteria complicate identification of individual projects; and c) there is no clear distinction between cognitive architectures and other intelligent systems, such as robot control architectures, expert systems, reasoning engines, frameworks, neural models, and large-scale brain simulations.

In this book, we identify projects primarily by their names. To honor the diversity of opinions in the literature, we considered all projects that were relevant to the declared purposes and goals of cognitive architectures, even if the authors of those projects did not explicitly consider them as such. After excluding frameworks and specialized models (e.g. standalone simulations of emotion or specific tasks), we estimate the total number of projects as not exceeding 250.

1.6 What architectures are covered in this book?

Our goal in writing this book was to focus on the significant projects that have broad lessons. Hence, we reduced the initial set of 250 architectures to a more manageable size by establishing the following inclusion criteria:

1. The architecture should have a minimum of 5 peer-reviewed publications;
2. There is sufficient evidence that the architecture has been implemented as a software or hardware artifact (e.g. source code, demo, etc.);
3. The publications should have at least 50 citations (this requirement was relaxed for more recent projects).

The first requirement sets some expectations for the maturity of the project. Given the complexity of the human mind, even projects focusing on isolated

[7]

aspects of cognition require significant time commitment. Most known architectures have been in development for decades.

The second requirement provides some guarantees that the theoretical concept is practically viable. There are, of course, many theoretical architectures that were not implemented by their authors but had significant influence on the field, e.g. Minsky's Society of Mind or Ullman's visual routines. We do not cover them separately, but rather through the prism of the implementations that they inspired. Another example is the Common Model of Cognition (CMC), which is an effort toward creating a standard model of the mind, much like a standard model in physics. It seeks unification of three successful cognitive architectures, ACT-R, Soar, and Sigma. From its origins from the 2013 AAAI Fall Symposium on Integrated Cognition (Burns et al., 2014), the CMC project generated many fruitful discussions, but its goal is to provide a blueprint and direction for further development of human-like rather than a concrete implementation (Laird et al., 2017), therefore we do not discuss it in this book.

The third requirement is to ensure that the work is impactful. Thus, we excluded architectures where more than half of the papers were uncited and those with too few citations overall.

After careful consideration, we selected a set of 68 cognitive architectures and 18 related projects (listed in Table 1.2) that have been influenced by a number of fields, including philosophy, psychology, cognitive science, neuroscience, biology, engineering, and AI. Approximately half of these projects are still being actively developed. This covers nearly half a century of research: work on the oldest architecture in this book, RCS, began in the early 1970s, whereas the most recent project we considered, MBCA, was introduced in 2018.

1.7 Summary

- The term "cognitive architecture" refers to a proposal for the design and implementation of an artificial entity capable of intelligent behavior in a reasonably complex environment. Cognitive architecture is akin to a blueprint which specifies the necessary components that enable a variety of abilities and that remains stable over the lifetime of the system. However, architectures are not immutable—both their contents and structure can gradually change over time.
- The number of modules within the system, their functionality, internal representations, etc., as well as information flow between the modules may differ significantly across architectures; however, they should be to some extent based on the understanding of human cognition and exhibit comparable behavior.
- Implementation in the form of software or physical artifact is generally expected to accompany the theoretical description, but is not mandatory. We limit the discussion in this book to cognitive architectures with implementations because we see it as a proof of practical viability.
- The relationship between the implementation and theory is not one-to-one: for any given theory there may be multiple possible algorithmic solutions

and a specific implementation can be interpreted from various theoretical standpoints.

- Theoretical backgrounds of cognitive architecture range from a complete unified proposal to a disjointed set of theories that do not contradict each other. Most architectures focus on modeling some aspects of cognition in detail and approximate or omit the rest.
- Cognitive architectures typically take years, if not decades, to develop, often evolving away from the initial proposal. These changes are sometimes formalized as a new version of the architecture or a new architecture altogether. However, versioning and naming conventions are not consistent across projects, because there are no agreed upon criteria for what amount of change calls for a new version of the architecture vs. a new architecture.
- For the reasons above, the exact count of cognitive architectures cannot be established, but we estimate that nearly 250 cognitive architectures have been proposed since the 1970s. The publications and implementations of 68 architectures and over a dozen of related projects were used as a reference for writing this book.

Table 1.2 Cognitive architectures covered in this book (in alphabetical order). "Timeline" for each architecture is established based on the publications or project webpages. "Present" means there was a publication or a project update within the last three years (since 2021). The "related projects" column lists significant extensions or predecessors of the respective architectures. These were considered when discussing theory and practical abilities of the architectures. "Key references," in our opinion, provide the best overview of the respective cognitive architecture.

	Timeline	Architecture	Related projects	Key references
1	1996–2007	3T	RAP	Bonasso et al., 1997
2	1976–present	ACT-R		Anderson et al., 2004; Ritter et al., 2019
3	2004–2017	ADAPT		Benjamin et al., 2004
4	1983–2003	AIS		Hayes-Roth et al., 1992; Hayes-Roth, 1995
5	2015–present	ARCADIA		Bridewell and Bello, 2016
6	1976–present	ART		Grossberg, 2021
7	1991–1994	ATLANTIS		Gat, 1998
8	1981–2007	BBD		Krichmar and Edelman, 2005; Edelman, 2007
9	2007–2018	BECCA		Rohrer, 2012
10	1981–2007	CAPS		Just and Carpenter, 1992; Varma, 2014a
11	2006–2016	CARACaS	CAMPOUT	Huntsberger et al., 2011
12	2009–2014	CERA-CRANIUM		Arrabales et al., 2009c
13	1992–present	CHREST	CHREST+	Gobet and Lane, 2012
14	1993–2008	CIRCA	Hy-CIRCA, SA-CIRCA	Musliner et al., 1993
15	1994–present	Clarion		Hélie and Sun, 2010; Sun, 2020
16	1993–2002	Cog		Brooks et al., 1999
17	1989–2019	COGNET		Zachary et al., 1998
18	2008–present	CogPrime	DeSTIN	Goertzel et al., 2013
19	2004–present	CoJACK	JACK	Ritter et al., 2012
20	2004–present	Companion		Forbus and Hinrichs, 2006
21	1984–2006	Copycat	Metacat	Hofstadter and Mitchell, 1994
22	2015–present	CORTEX	RoboCog	Bustos et al., 2019
23	1992–present	DAC	DAC-h1, DAC-h3	Verschure, 2012
24	2005–present	DIARC		Schermerhorn et al., 2006; Scheutz et al., 2019
25	1986–present	Disciple		Tecuci and Hieb, 1996; Tecuci et al., 2019
26	1994–2020	DUAL	AMBR	Kokinov, 1994b
27	1994–present	EPIC		Kieras, 2007
28	2010–present	ERA		Morse et al., 2010
29	1990–1993	ERE		Bresina and Drummond, 1990
30	1991–present	FORR		Epstein et al., 2002
31	1992–2013	GLAIR	MGLAIR	Shapiro and Ismail, 2003
32	2007–present	HCA		Haikonen, 2007
33	1989–present	ICARUS	PUG	Choi and Langley, 2018
34	1990–present	IMA		Kawamura, 2023
35	1995–present	IMPRINT		Mitchell, 2000; Allender, 2000
36	1998–2009	Kismet		Breazeal, 2003a
37	1996–present	Leabra		O'Reilly et al., 2012
38	2006–present	LIDA	IDA, VMattie, CMattie	Baars and Franklin, 2003; Franklin et al., 2016
39	1997–present	LISA	DORA, JIM, MoralLISA	Hummel and Holyoak, 2003
40	1999–present	MAMID		Hudlicka, 2002
41	2018–2021	MBCA		Schneider, 2018
42	2001–present	MDB		Bellas et al., 2010a
43	2017–present	MECA		Gudwin et al., 2017
44	1991–2019	Meta-AQUA		Cox, 2005
45	1981–present	MHP	MHP-RT	Card et al., 1986
46	2003–2019	MicroPsi	Psi	Bach, 2009
47	1985–2019	MIDAS	APEX	Freed, 1998
48	2011–present	MIDCA	Meta-AQUA	Cox et al., 2022
49	1983–present	NARS		Wang, 1995a; Wang et al., 2016; Wang, 2022
50	1993–2014	OMAR	D-OMAR	Deutsch and Pew, 2019
51	1986–2011	OSCAR		Pollock, 1987
52	2002–2013	Polyscheme		Cassimatis et al., 2009
53	1986–2005	PRODIGY		Veloso et al., 1995
54	1986–2004	PRS		Georgeff and Lansky, 1987; d'Inverno et al., 1998
55	1988–1992	RALPH		Russell, 1991
56	1971–2011	RCS	4D/RCS, NASREM	Albus, 1991
57	2008–2014	SAL		Jilk et al., 2008
58	1993–2008	Saphira		Konolige et al., 1997
59	2008–2014	SASE		Weng, 2002
60	1993–2013	SHRUTI		Shastri, 1999
61	2008–present	Sigma		Rosenbloom et al., 2016
62	1982–present	Soar		Laird, 2012a; Laird, 2022a
63	2012–present	SPA		Eliasmith et al., 2012; Eliasmith, 2013
64	2006–2019	SS-RICS		Kelley, 2006
65	2014–present	STAR		Tsotsos et al., 1995; Tsotsos and Kruijne, 2014
66	1985–2004	Subsumption		Brooks, 1986; Brooks, 1990
67	1989–2002	TCA		Simmons, 1994b; Simmons, 1994a
68	1995–2013	Ymir		Thórisson, 1998

2 Cognitive Architectures, AI, and Cognitive Science

Cognitive architectures evolved at the intersection of artificial intelligence (AI) and cognitive science and absorbed concepts and methods from both disciplines. But what makes cognitive architectures unique is the pursuit of a holistic understanding and implementation of the human mind. As such, they can test empirically what mechanisms are consistent with psychological evidence and result in intelligent behavior, and which are beneficial for advancing theory and applied research.

In this chapter we will discuss how breakthroughs in algorithmic techniques and understanding the human mind were reflected in the development of cognitive architectures. Even though some of the earliest attempts to build intelligent machines date as far back as antiquity, we will limit the overview to the computer era that started in the 1950s. Since then, every decade brought increasing amounts of empirical data and algorithmic innovation. The field of cognitive architectures was growing in parallel. From the 1970s until now, almost every year at least one new cognitive architecture has been proposed. By 2010 nearly 60 architectures were being developed concurrently.

Section 2.1 provides a historical context by describing significant events and trends in the development of AI and cognitive science.

Section 2.2 overviews a number of theories of cognition and algorithms which influenced cognitive architectures.

Section 2.3 discusses mutual influences among the three disciplines.

2.1 Historical context

Creating artificial beings that can think and act like humans is not a new idea. In some form or another, it has been pursued by humanity for ages with various motivations: to learn more about ourselves, to eliminate menial work, or to achieve the ability to create new life. But a rigorous scientific investigation into the computational nature of the human mind and the possibility of recreating it began only about a century ago.

Most sources point to a series of events in mid-20th century that led to the formation of new fields of artificial intelligence (AI) and cognitive science, both of which had a direct influence on the development of cognitive architectures. Turing (1937) showed how rational behavior can be produced by a relatively simple computational agent—the Turing machine. McCulloch and Pitts's (1943) designed a neuron model with simple linear thresholding units that could perform the same computations as Turing machines; this was later formally proved by Arbib (1961). Wiener

of feedback control that allowed the agents to learn from their experiences and adapt their behavior. The now standard von Neumann computer architecture proposed in 1945 outlined major functions (central arithmetic part, central control part, memory, input, output, and external memory) and their implementation via E-elements inspired by the McCulloch-Pitts neuron model (von Neumann, 1993).

During this time, the human mind/brain was increasingly viewed in computational terms, promising to solve its mystery. Turing machines explained how information processing might occur in such a system, neural networks based on McCulloch-Pitts neurons linked neural computation and observed behavior, and von Neumann architecture made their physical realization possible.

2.1.1 Artificial intelligence

Artificially intelligent entities have been portrayed in fiction long before any such thing existed in reality. The modern concept of human-like machines began with Karel Capek's play R.U.R (Rossum's Universal Robotics) published in 1920. The word "robot" in the title was derived from the Czech "robota," meaning "drudgery" or "servitude," and originally referred to what we now call androids, as robots in the play were indistinguishable from humans. Only some years later, Isaac Asimov in his popular Robot Series introduced the word "robot" into English and formulated the famous three laws of robotics. From then on, countless science fiction books and films envisioned various futures of AI and robots, inspiring public imagination and generations of scientists, and reflecting the hopes (and, sometimes, fears) of what advanced AI could bring.

One of the first non-fiction works on the topic of machine intelligence is the seminal paper "Computing Machinery and Intelligence" by Turing (1950) who argued for the feasibility of computers behaving intelligently, formulated an evaluation procedure for assessment of intelligence (aka the Turing test, see Chapter 10), suggested a developmental approach to constructing thinking machines, and proposed some initial tasks worth pursuing: playing chess and natural language understanding.

The term "artificial intelligence" appeared several years later in a proposal for a summer research project to be carried out at Dartmouth College in 1956 (McCarthy et al., 2006). The proposal described a two-month ten-man project to explore natural language comprehension, knowledge representation, self-improvement, forming abstractions from sensory inputs, and creative thinking. Later dubbed the Dartmouth Conference (Moor, 2006), this summer workshop ran for several weeks and was attended by an illustrious group of participants, including Marvin Minsky, John McCarthy, Claude Shannon, Oliver Selfridge, Herbert Simon, and Allen Newell (according to Solomonoff, 1956).

Needless to say, the lofty goals set seventy years ago have not been reached yet. In retrospect it is obvious that both the failure of the Dartmouth project and slow progress toward goals set by Turing have largely resulted from the mistaken assumption that enough was understood about human cognition to encode it computationally. Nevertheless, these setbacks not only did not hamper, but stimulated research in AI and shaped its progress for many decades.

Early years (1950–1970s). The first years following the establishing of AI as a discipline were fueled by the early successes and expectations of truly

intelligent machines within the next few decades. During this time, the focus was mainly on representation and search with many successful applications of AI appearing in narrow domains like chess and mathematical theory-proving, where expert knowledge combined with formal logical manipulations could quickly arrive at the solution. Among the famous achievements of this period are:

- Logic Theorist—a program that reproduced dozens of proofs from *Principia Mathematica* (Newell et al., 1958);
- General Problem Solver (GPS)—a successor of Logic Theorist for solving problems in logic, trigonometry, and planning (Newell et al., 1959);
- STUDENT—a program for solving high-school algebra problems (Bobrow, 1964);
- ELIZA—the first chatbot, primarily known as an AI psychotherapist[1] (Weizenbaum, 1966);
- Teachable Language Comprehender—a program for English text comprehension (Quillian, 1967);
- DENDRAL—a program for inferring the structural hypotheses of molecules based on the analysis of mass spectral data (Buchanan et al., 1969);
- MYCIN—a program for medical diagnosis and therapy (Shortliffe et al., 1973);
- SHRDLU—a program for conversing about objects in the simulated Blocks World and their properties (Winograd, 1980).

Another invention of that era was the Lisp programming language, in which all but two of the projects above were written (Logic Theorist and ELIZA were written in IPL and MAD-SLIP, respectively). Created in 1958 by McCarthy, Lisp supported applying predicate logic to reasoning tasks, from answering questions to chess-playing, and remained a staple programming language for most research in AI and cognitive science for several decades (McCarthy, 1978).

Even though domains of early AI experiments were limited to micro-worlds of simple objects and words, the effectiveness of symbolic representations was well demonstrated. Following initial successes, symbolism was strongly endorsed by many influential figures in the field as *the* way of achieving general intelligence in the future.

First AI winter in late 1970s. The 1970s were marked by the first "AI winter" associated with decline in interest and funding for AI projects due to a combination of overly optimistic projections, unfulfilled promises, and technological barriers, such as limited computing power, memory, and slow processing speeds. The shortcomings were summarized in a series of reports: a report by Automatic Language Processing Advisory Committee (ALPAC) commissioned by the American government (ALPAC, 1966), a report written by Sir James Lighthill (1973) at the request of the British government, and, lastly, a report by an American Study Group (ASG) assembled by the USA Department of Defense to assess DARPA's AI program (as noted in the report by National Research Council, 1999).

[1]This behavior was enabled by the DOCTOR script which imitated responses of a Rogerian psychotherapist. Other scripts existed, including in other languages, such as Welsh and German. The therapist version of ELIZA is still "alive" on most computers that have the Emacs text editor. To start a session, launch Emacs from the terminal using command `emacs -nw` and enter `M-x doctor` (press the Escape key, `X`, and type `doctor`) inside the Emacs

The ALPAC report focused on machine translation from foreign languages into English. It reached the conclusion that significant government funding did not result in the prospect of usable machine translation in the foreseeable future. While raw machine output was decipherable, it was generally difficult to read and contained errors that obscured or modified the meaning of the original, e.g. unnatural word order, transliteration of unknown words, and incorrect or unusual translations of common words.

The Lighthill report looked broadly at all major areas of AI (category A), robotics (category B), and computational cognitive science and neuroscience (termed category C for study of the central nervous system). In research categories A and C, respectable achievements were acknowledged, but only in narrowly defined domains and to a smaller extent than had been hoped. Progress in robotics, however, was singled out as the most disappointing, partly because of higher initial expectations, but also due to issues with general problem-solving and common-sense reasoning, in addition to a lack of adequate motor coordination, needed for automating everyday and industry tasks. Even advances in specific domains were not deemed sufficient. For example, early research in AI pursued solving chess-playing and other tasks that were difficult for most humans, assuming that the results would generalize to other human activities that involved reasoning and problem-solving. However, programs for playing chess in 1970 had already achieved the level of human amateurs through clever heuristics that had no feasible uses in other areas and no clear path toward general intelligence.

The common cause for slow progress was identified by Lighthill as the exponential combinatorial growth of the search spaces that represented possible solution paths that existing methods could not deal with effectively. These disappointing findings were seconded by the ASG group and shortly after led to cuts in AI funding in universities across Europe and North America, precipitating what has become known as the first AI winter.

Brief resurgence of AI (1980s). In the late 1970s and early 1980s, a number of knowledge-based systems from a decade ago were commercialized, many by the academics who developed them. Thus began a new industry of AI expert systems (Vaux, 2001). These domain-specific programs used extensive knowledge bases and inference procedures to solve problems at the human expert level. A large role in this endeavor is attributed to the Fifth Generation Funding Program (FGFP). It was initiated by the Japanese government in 1982 and provided funding for research on parallel computing, logic programming, and specialized hardware, which further boosted expert systems (Moto-Oka and Stone, 1984). The US government across its many agencies also invested millions in expert systems research (Gevarter, 1982).

As a result, by the mid-1980s, there were more than sixty commercial expert systems deployed in various sectors. Besides well-known examples, such as DENDRAL for chemical analysis, XCON for circuit design, and PUFF for medical diagnosis, expert systems were developed for agriculture, education, manufacturing, software development, and military purposes (Buchanan, 1986). The revenues of commercial AI applications went from a few million dollars in the early 1980s to hundreds of millions by the mid-1980s (Szolovits, 1989). At the same time, new industries formed around building workstations optimized for running code written in Lisp and Prolog that remained as the

go-to languages for programming AI and designing development environments for creating expert systems.

Once again, the potential of the new technology turned out to be exaggerated. While logic programming was applicable in many areas, from medicine to law, the fundamental issues with learning and understanding natural language were not resolved. Expert systems were difficult to update, did not have common sense, and operated under the "closed world" assumption, meaning that what was not known to be true was presumed to be false (Bell, 1985). In the commercial setting, expert systems proved to be unreliable, hard to program, and expensive to maintain, which limited their economic utility.

By 1995, only about one-third of expert systems (out of a hundred systems surveyed) were still in use (Gill, 1995). A final blow was dealt by the improvements in general computing hardware and the emergence of procedural and object-oriented programming languages that enabled the use of standard von Neumann computers for the development of AI instead of specialized Lisp workstations (Myers Jr and Yamakoshi, 2020). While Lisp remained the dominant programming language for AI, the market for Lisp machines collapsed within a few years (Graylin et al., 1998).

Second AI winter (late 1980s). A new wave of disappointment with the progress in AI arrived in the late 1980s. Government funding once again dried up, and commercialization of AI waned, precipitating the second AI winter. According to Hendler (2008), AI never truly recovered from the funding reductions caused by the first AI winter ten years prior. For a number of years, the amount of fundamental research coming from the universities dwindled and even though industry investments compensated for receding government funding, they encouraged researchers to pursue low-hanging fruit instead of tackling deeper issues. Eventually, lack of funding and innovation, together with growing criticisms of the approaches to AI within the research community, led to cooling attitudes about the future of the discipline.

Despite unfavorable circumstances, several notable discoveries were made during this time. Particularly, interest in neural networks began to gain momentum again in the 1980s, spurred by the efforts of Feldman and Ballard (1982) to legitimize connectionism and discovery of new methods for training multilayer perceptrons (Pollack, 1989). In particular, learning via backpropagation was quickly established as a key feature of connectionism since it provided a simple and effective solution for the long-standing "credit assignment problem".[2] Important advancements were made in reinforcement learning (RL) as well. Although the first works on RL appeared in the late 1950s, it was not an active research area until 1980s (Sutton, 1992) when Sutton

[2]Because backpropagation became the backbone of many recent achievements in AI, a different kind of credit assignment problem arose—who came up with it first. According to Dreyfus (1990) and Boden (2006), the idea of gradient propagation was developed independently in the 1960s by Kelley (1960) and Bryson and Denham (1962) for optimal control theory. Werbos (1974) was the first to apply backpropagation to neural networks in a social science domain, but the approach did not gain traction until it was rediscovered by Parker (1982) and Rumelhart et al. (1986b) a decade later. Nevertheless, the Parallel Distributed Processing (PDP) research group headed by Rumelhart and McClelland presented backpropagation as a novel development in the eponymous volume (Rumelhart et al., 1986b) and seminal *Nature* paper (Rumelhart et al., 1986a). Later books, such as the one by Chauvin and Rumelhart (1995) and a popular deep learning textbook by Goodfellow et al. (2016), acknowledged the earlier discoveries of backpropagation but continued to name PDP group's work as the primary source.

and Barto (1987) introduced the temporal difference (TD) model of classical conditioning and Watkins (1989) proved convergence of Q-learning.

AI in the shadows (late 1990s). After two consecutive AI winters in the 70s and 80s, the next decade saw diminished activity in what people called AI. But research did not stop and eventually gave rise to significant developments in search algorithms, constraint solvers, intelligent agents, and operations research (Nield, 2019). Many important practical advances were achieved during this time. Among them is TD-Gammon, a backgammon program that applied temporal-difference (TD) reinforcement learning to play on par with top human players (Tesauro, 1991), as well as Support Vector Machines (SVM) (Vapnik, 1995) and random forests (Breiman, 2001), that became staple methods for pattern recognition.

Concurrently, early successful applications of neural networks to real-world problems also took place. In particular, LeCun et al. (1989) applied backpropagation to convolutional neural networks (CNN), first featured in Fukushima's (1980) Neocognitron, to perform the task of automatic recognition of handwritten ZIP code digits. The data was provided by the US Postal Service and eventually became part of one of the earliest image datasets called MNIST (LeCun and Cortes, 1998). In parallel, recurrent neural networks were developed by Hochreiter and Schmidhuber (1997).

Perhaps, the most publicized milestone in the development of AI of that time was the defeat of the world chess champion Garry Kasparov by IBM's Deep Blue in 1997. The key to the machine's win was a massively parallel search engine powered by an extensive game database and specialized hardware. As a result, Deep Blue could search for optimal moves through 100 to 200 million chess positions per second in its enormous database of hundreds of thousands of chess matches (Campbell et al., 2002).

Comeback of AI (2000s to late 2010s). The new rise of AI in the mid-2000s was marked by a number of events. In 2004 and 2007, DARPA sponsored and organized two competitions to test the viability of autonomous driving off-road (Grand Challenge) and in the city (Urban Challenge), thus initiating the race toward self-driving vehicles. Shortly after, the age of large-scale datasets that fuel modern research began with the introduction of ImageNet in 2009 (Deng et al., 2009) that contained over a million human-annotated images. In 2012, a deep CNN called AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky et al., 2012). Wide availability of graphic processing units (GPUs) repurposed for general computation and cheap memory storage also played a role; at last, combinatorics of training large neural networks (Froese and Hertrich, 2023) could be brute-forced to a useful extent.

Although "classical" statistical machine learning methods continued to demonstrate impressive results—in 2011 IBM Watson DeepQA defeated human champions in the popular TV quiz show Jeopardy (Ferrucci et al., 2010), deep learning quickly made them obsolete and soon became synonymous with AI. In mid-2010s, deep learning was expanded to RL (Mnih et al., 2015) and generative modeling (Goodfellow et al., 2014). Shortly after, Transformer models were introduced (Vaswani et al., 2017). Together, these discoveries form the basis for the current mainstream AI dominated by foundation models, large language models, and chatbots that rely mainly on scaling data and

compute to perform numerous tasks and can communicate in multiple modalities (Bommasani et al., 2021). The most famous of these is ChatGPT, which amassed millions of users months after its release by OpenAI in November 2023 and kickstarted the wave of generative AI.

The levels of investment and hype around AI that we are seeing now (in 2025) are unparalleled in their scale but reminiscent of the past in other aspects. Just before the second AI winter set, a panel discussion held at AAAI-84 raised many issues that will sound eerily familiar to anyone involved in modern AI. For example, the panelists discussed how industry participation led to prioritization of short-term goals and financial gains over fundamental research, replacement of scientific method with software engineering in academic publications, top academic conferences doubling as job fairs, and the overhyped potential of AI coupled with exaggerated fears of its dangers (McDermott et al., 1985).

Forty years later, AI researchers voice similar concerns, according to a recent survey (Su and Crandall, 2021). Other new and troubling trends are also apparent, such as "celebritization" of a small group of academics, marginalization of non-deep-learning research in AI, dismissal of prior work, muddling of theory, and rapid growth of low-quality literature due to lowered barriers for entry.

Few will argue that we live in a world where AI is again seen as a major driver of scientific and technological progress. Today, deep learning has practically monopolized academic research in many disciplines and is being adopted in many products, services, and applications. But it is yet to be determined whether the trajectory we are on will lead to the transformational outcomes being advertised. After all, deep learning may not in fact be a path to human-level intelligence (Marcus, 2018). Recent failures of some AI tools to deliver on the promises, such as the slow progress of self-driving cars and closure of IBM's Watson project, hint at such a possibility (Raji et al., 2022). Even the financial benefits of adopting AI in business settings for automating tasks or augmenting workers are not proven yet (Enholm et al., 2022; Nathan et al., 2024). Should technical roadblocks and failure to recoup the investments cause another AI winter, it is likely to be disastrous for the field, given the enormous resources and efforts vested in AI.

2.1.2 Cognitive science

The beginnings of cognitive science in the 1950s are referred to in the literature as a "cognitive revolution" characterized by the rejection of behaviorism that dominated psychology at the time (Amsel, 1992). In short, behaviorism, primarily associated with the works of Skinner (1977), posits that behavior is the subject of scientific enquiry. As such, it does not study that which cannot be directly observed, i.e. mental states that may cause the behavior, and instead focuses on functional relations between behavior and environmental variables (Moore, 2011).

The original goal of behaviorism was to show that any behavior can be represented as a reflex by finding the corresponding conditioned behaviors and operants (Malone, 1975), but it was becoming clear that the explanatory power of conditioning had been overestimated. In addition to failure of behaviorism

to account for higher mental abilities, several other factors contributed to its demise, which, according to Mueller and Mueller (1995), include:

- Chomsky's (1959) research in linguistics that contradicted the behaviorist explanation for incremental language acquisition presented by Skinner (1957) and demonstrated strong evidence for newborns' abilities to master language without formal training and with sporadic reinforcement;
- Piaget's (1952) theory of cognitive development that claimed assimilation and accommodation rather than reinforcement as key to developing intelligence in children;
- Emergence of information-processing models in computer science and information theory that provided useful tools for describing and explaining cognitive processes.

As unobservable cognitive processes were opened up for investigation, "cognition is computation" became the motto of the new discipline called cognitive science that combined computational modeling with findings from psychology, and later neuroscience, to understand how the human mind processes information. Cognitive science, born mere months after AI, was influenced by the same pioneering works in information theory, psychology, and computer science discussed in the previous section, and followed a similar path in the beginning.

The beginning of cognitive science is dated to the September 11th, 1956, the second day of a symposium organized by the "Special Interest Group in Information Theory" at MIT. The symposium took place shortly after the Dartmouth summer school and was attended by many of the same people (Miller, 2003). On that day, Newell, Shaw, and Simon presented their General Problem Solver (Newell et al., 1958), and Chomsky and Miller presented their works in theoretical linguistics (Chomsky, 1956) and memory bottlenecks (Miller, 1956). Shortly after, in 1958, another influential event, the Teddington symposium, was held in the National Physical Laboratory (NPL). Works presented there defined future trajectories of cognitive science and AI. Among them were Selfridge's (1959) Pandemonium, Rosenblatt's (1958) Perceptron, and McCarthy's (1959) proposal of imbuing programs with common sense.

Early years (from 1950s to 1970s). Initially, cognitive science followed the path of AI research by focusing primarily on knowledge representation and high-level cognitive processing, such as thinking, planning, and problem-solving. In particular, much of the effort was concentrated on investigating explicit forms of knowledge representation, such as symbols and rules, or other intermediate forms, for example, Quillian's (1967) semantic networks.

In parallel, an active area of memory research examined aspects of human sensory, short-term, and long-term memories, as well as different types of knowledge, such as procedural, declarative, and episodic. The processes that operated on symbols stored in memory were related to mental activities: *accessing* was seen as memory retrieval, *transformation* as problem-solving, and *alignment* as analogy (Gentner, 2010).

Rise of connectionism (1980s). The initial focus on symbolic processing led to many promising results in theorem-proving and chess-playing, but did not translate well to demands of real-world tasks. Connectionism, despite being developed in parallel with symbolism, was not considered a viable alternative

for quite some time. Most accounts point to Minsky and Papert's (1969) critical assessment of perceptrons showing that they could not solve many problems (connectedness, parity, etc.) as the culprit, discouraging others from working on neural networks and diverting funding away from that line of research.

Connectionism did not die out entirely, since many researchers, including Minsky himself, continued to work on both approaches. A number of events helped it recover. First, Hofstader's (1979) book *Gödel, Escher, Bach: An Eternal Golden Braid* raised interest toward connectionism through discussions of music, logic, and biology. Scientific methodology was then established by Feldman and Ballard (1982), who advocated for broader use of connectionist models in cognitive science and demonstrated their advantages for tackling issues in perception and motor control, as well as their potential for reasoning and problem-solving. Finally, publications from PDP research group in 1986 contributed to the further ascent of connectionism by disproving some of the earlier limitations as characteristic of one-layer perceptrons but not applicable to networks with several intermediate layers (Rumelhart et al., 1986b, p. 111). Some viewed the shift to connectionism as a return of behaviorism and empiricism that prompted a cognitive revolution earlier (Papert, 1988; Place, 1992).

Expansion of cognitive science (1990–present). These several decades have been characterized by downward and outward expansion of cognitive science (Frankish and Ramsey, 2012) and divergence of cognitive science from research in AI. On the one hand, new technologies, such as functional magnetic resonance imaging (fMRI) provided new views of many cognitive processes and allowed further integration with neuroscience. On the other hand, phenomena, such as emotion, consciousness, non-human cognition, and embodiment were the emerging countertrends to mainstream cognitive science. New approaches saw cognition as not something confined to one's head but rather as situated (context sensitive), temporal (varying according to time availability), distributed (cognitive functions are not localized to one brain area and some may be off-loaded onto the environment), action oriented, and embodied (body defines perception) (Larkin et al., 2011). The latter was influenced by Gibson's work on the ecological approach to perception, which emphasized that much of the information was perceived directly, not processed step by step. Barsalou (1999) developed perceptual symbol systems as embodied systems. In robotics, Brooks (1987) demonstrated that internal representations are not necessary for generating some simple behaviors.

From the outset, cognitive science was meant to be a diverse field comprised of psychology, linguistics, AI, anthropology, philosophy, and neuroscience. The idea was that through interdisciplinary collaboration, the founding disciplines would eventually merge. In the meantime, the credibility of findings would improve via critical examination from different viewpoints.

Whether this goal has been achieved remains a debated topic. Early analyses of bibliometric patterns and author affiliations in the premier cognitive science journals showed that neither merging of contributing disciplines nor balance of their contributions was attained. Numerous studies of bibliometric data and author affiliations showed that in less than twenty years upon its founding the interdisciplinarity of the *Cognitive Science* journal and the field

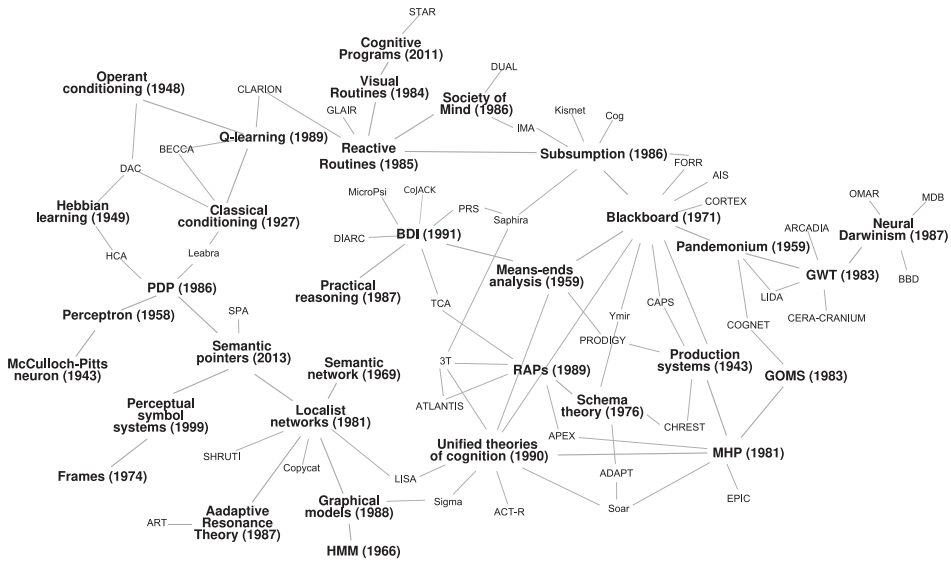

Fig. 2.1 Theoretical roots of cognitive architectures. Important theories and models are shown in bold font, with date of introduction in brackets. Cognitive architectures are shown in regular font. Edges show connections between theories and architectures.

has decreased significantly and continued on the downward trend with evidence of being overtaken by psychology (Schunn et al., 1998; Shin, 1999; Goldstone and Leydesdorff, 2006; Gentner, 2010; Leydesdorff and Goldstone, 2014). Particularly, Núñez et al. (2019) argued recently that cognitive science has been largely absorbed by psychology and lost its influence on neuroscience, anthropology, and philosophy. The authors' verdict of "failed interdisciplinary integration" in cognitive science spurred a flurry of responses (Núñez et al., 2020), some of which reached an opposite conclusion by looking at different journals (Bender, 2019) or using different measures for analysis (Alasehir and Acarturk, 2022; Contreras Kallens et al., 2022).

2.2 Roots of cognitive architectures

As many of the same people who established cognitive science and AI also participated in developing cognitive architectures later, this field was naturally influenced by theories of cognition and computational solutions proposed in both disciplines. We identified a handful of the most important theories and models by tracing common references from individual architectures and plotting them as a graph in Figure 2.1. Below we discuss contributions of the central works.

Means-ends analysis. Means-ends analysis was developed in the early days of AI by Newell et al. (1959) for the GPS. "Means-ends" is conceptually simple: it approaches problem-solving as building action sequences given the specifications for possible actions, the initial situation, and the goal description. For example, GPS reached the goal from the initial state by finding operators that reduced the differences between the current and desired states (also known as

forward chaining) (Newell and Simon, 1961), a heuristic that has been observed in human problem-solving behavior (Newell and Simon, 1972).

This method later was popularized and extended in STRIPS (Stanford Research Institute Problem Solver) (Fikes and Nilsson, 1971) and a family of PRODIGY planning algorithms (Fink and Blythe, 1998). Further development of means-ends analysis within the planning community led to large improvements in performance but arguably pushed it to be less human-like, e.g. by replacing forward chaining with backward chaining and pursuing bidirectional approaches (Fink and Blythe, 1998) with increasingly high memory requirements for storing partial plans, and use of constraint satisfaction techniques (Langley, 2006b).

Means-ends analysis had a lasting influence and continued to be explored both in cognitively-inspired architectures, such as ACT-R, ICARUS, Soar, and GLAIR, as well as in more engineering-focused applications within CIRCA, ERE, and BDI-based architectures.

Blackboard. The concept of *blackboard* architecture also belongs to Newell (1962), who described it metaphorically as a set of workers looking at the same blackboard, each being able to read from it and judge whether they can add anything to it within their area of expertise. The idea arose as a solution to the communication problem between subroutines within the GPS mentioned earlier. Blackboard also bears a resemblance to Selfridge's (1959) Pandemonium, where demons independently assess the solution and respond depending on how well it fits their natures. Newell himself went on to work on the production rule system (PSG) (Newell, 1973a). Meanwhile, according to Nii (1986), Simon suggested the blackboard analogy to the HEARSAY team, who developed its first complete implementation for the HEARSAY-II speech understanding system (Erman et al., 1980).

The basic blackboard architecture has three essential components: the blackboard itself, knowledge sources (experts), and a scheduler (Schwartz, 1995). Experts read from the blackboard and determine items to be added to the blackboard without consulting one another. The role of the scheduler is to select the most appropriate knowledge sources to contribute to the blackboard. Various modifications were added in subsequent implementations, such as changes to the structure of the blackboard (e.g. hierarchies, partitions, and contexts), representations (shared across the entire blackboard or separate ones for each partition), and scheduling (knowledge or event oriented) (Craig, 1988). Naturally, division of labor across many experts offers many opportunities for parallelism, such as concurrent updates to the blackboard, parallel execution of knowledge sources, distributed control, and multiple instances of knowledge sources exploring alternate hypotheses or using different inputs (Schwartz, 1995).

The conceptual clarity and versatility of the blackboard approach made it a popular choice for many cognitive architectures, including BB1 (a descendant of HEARSAY-II), Copycat, CAPS, FORR, Ymir, and, more recently, CORTEX.

Production systems. Production systems were initially proposed by Post (1943) as a general computational mechanism. In its purest form, a production system consists of a finite set of productions (formulated as if-then rules), a database represented by a collection of symbols on which the rules operate, and

an interpreter for the rules (Davis and King, 1984). Production systems were widely used throughout the 1960s and 1970s in models of human cognition and in expert systems.

Production systems, adopted as a formalism for describing human problem-solving behavior by Newell in the 1970s, retained the components of the original blackboard but reinterpreted their meaning and function. The database became the contents of working memory with limited capacity, and symbols within it became chunks of the knowledge following Miller's (1956) memory theory. The main difference between the two was finer granularity of productions compared to knowledge sources that could be arbitrarily complex (Pfleger and Hayes-Roth, 1997). During execution, productions whose conditions matched elements residing in working memory at the time were executed and modified working memory, in turn, triggering execution of other productions. This cycle repeated until the desired goal was satisfied.

Due to the central role of memory, initial applications of production systems in psychology aimed to replicate human memory tasks. By manipulating sets of productions, different behaviors and biases observed in human data could be simulated (Davis and King, 1975). Thus, production systems became a basis of many well-known cognitive architectures, including Soar, ACT-R, CAPS, EPIC, CHREST, SS-RICS, and Clarion.

Society of Mind (SoM). The concept of SoM was initially developed by Minsky together with Papert in the 1970s but was published in a book a decade later (Minsky, 1987). The new theory offered several significant updates to the existing ideas about modeling human cognition, particularly in areas of language, memory, and learning. It presented the mind as a product of activity of many diverse specialized agents performing cognitive subprocesses. Each agent is a computer program that can be composed into larger systems, called societies of agents, capable of performing more than what a single agent could. SoM itself is a total set of all agents comprising the mind. In this system, mental activity reduces to turning individual agents on and off.

While superficially similar to the blackboard and production systems, SoM differs from both. It expects that hundreds, thousands, or even millions of agents may be needed to simulate a full range of human cognition, which is orders of magnitude larger than any production or blackboard system ever created. Neither of them offer a solution to resolving conflicts between agents on such a scale.

In one aspect, SoM is also an exact opposite of Newell's idea of the unified theories of cognition. Unlike Newell, who emphasized the need for finding the minimum set of basic mechanisms and representations for cognition, Minsky argued that the space of all cognitive processes is so broad that finding such a set is impossible—some algorithms will naturally be more suited for certain representations and impossible or hard to adapt to others. Thus, the question Minsky posed was not how to find a unified solution for all cognitive phenomena, but rather how to make multiple distinct processes communicate and cooperate to produce intelligence.

Although SoM has never been implemented as an architecture, it nevertheless had significant influence and inspired several successful architectures, among them SHRUTI and Polyscheme.

Global Workspace Theory (GWT). The idea of global workspace was proposed by Baars (1983) as a mechanistic account of information processing in the brain characterized by contrasting conscious and unconscious features of human cognition (Edelman, 1987). The theory postulates that the brain is a collection of distributed, highly specialized networks or modules that compete for access to the global workspace. Modules require resources (mostly information) to function and produce resources as output. To gain access to information, they have two options: exchanging information between one another by forming coalitions or via the global workspace that lets them broadcast their requirements and results of computation to all other processors beyond the local coalition. Only one coalition is allowed to access the global workspace at a time, and the selection is determined by competition that depends on the activation levels of individual processes and of the entire coalition. Only the global workspace is associated with consciousness, the processes limited to local coalitions are considered unconscious.

Conceptually, GWT is similar to the blackboard systems discussed earlier. The shared global workspace is the blackboard, whereas modules act as knowledge sources and attention performs the function of the scheduler (Franklin et al., 2013).

Based on Baars's theory Franklin and Graesser (1999) developed a software agent CMattie that borrowed elements of Selfridge's Pandemonium and Jackson's extensions to it, Copycat (Hofstadter, 1994), and Maes's (1989) behavior networks. Later, this work has been extended to a full-fledged systems-level cognitive architecture, LIDA (Franklin and Patterson Jr, 2006). Other architectures influenced by GWT include CERA-CRANIUM and ARCADIA.

Behavior-based robotics. Until the mid-1980s, most robot control systems followed the sense-plan-act approach, where three main components were perception, planning, and execution. Information about the environment picked up by the sensors was used to compose or update the world model. Goals and world model were then fed into a planner, which produced a sequence of actions to achieve the goal to be passed to effectors (Gat, 1998).

Planning was considered the most crucial component and received the most attention. However, any reasonably complex environment quickly revealed the weakness of deliberation. Planning took time, and as the robot contemplated what to do next, the surrounding situation changed, either leading to wrong actions or invalidating the plans (McDermott, 1992). One way to address the inadequacies of the traditional way of building agents was to challenge its assumptions. The new crop of approaches, collectively referred to as behavior-based robotics, did not rely on symbolic decision-making and maintained that behavior was a result of interaction between the robot and the environment.

Perhaps, the most radical and the most known departure from sense-plan-act principle is the Subsumption architecture proposed by Brooks (1986). Instead of functional (or vertical) division into sensing, deliberation, and execution, he advocated for task-based decomposition, also called behavioral or horizontal, where each layer or behavior was independent of others and could perform one simple task, e.g. changing direction when encountering an obstacle, wandering about the environment, or recognizing objects. These tasks were hard-coded as finite state machines and arranged in a hierarchy, with simpler tasks at the bottom and wires connecting their outputs to inputs

of the tasks above. Modules for each task all run in parallel and could be reset, inhibited (temporarily disabled), or subsumed by other modules. This architecture did not require any memory or internal state, relying on the "world as its own best model" instead (Brooks and Flynn, 1989).

In practice, Subsumption was highly effective for simple reactive tasks, such as navigation around obstacles, but it did not generalize well to more complex behaviors. The most complex of any Subsumption-based robots, Herbert, was designed to pick up soda cans from the office desks but according to some accounts was unreliable and had not completed the task flawlessly (Gat, 1998). Against the core principles of Subsumption, it also had to include an internal state to be able to retrace the steps back to its original location (Connell, 1989).

Other issues stemmed from the fixed implementation and ordering of the behaviors. As Hartley and Pipitone (1991) note, designing layers was incredibly difficult because small changes in the upper layers triggered a cascade of modifications in the layers below. Furthermore, determining the order of behaviors was not intuitive, as it could be different depending on the context and task at hand. Lack of internal state presented its challenges as well. An entirely reactive architecture had no way of maintaining long-term goals, or have a reliable picture of the world. As a result, without the internal state that could maintain scene persistence, objects around the robot would flicker in and out of existence due to inaccurate sensor readings, reflections, and occlusions.

Although pure Subsumption architecture turned out to be impractical, it inspired many new approaches that addressed some of its shortcomings. The lack of flexibility in ordering and selecting behaviors spurred a new line of research on dynamic action selection mechanisms, such as Maes's (1991) behavior networks, Motor Schemas (Arkin, 1989), Reactive Action Packages (RAPs) (Firby, 1989), and Agre and Chapman's (1990) plan-as-communication approach. Independently, Ullman (1983) proposed conceptually similar visual routines that explained how elemental visual operators and control structures are combined to perform complex visual tasks.

These approaches have been incorporated in many successful architectures. In the beginning of the 1990s, three groups worked independently on robot architectures AAA, 3T, and ATLANTIS that combined reactive behaviors with planning. All three had similar components: planner, reactive control, and a scheduler that connected the two together. AAA used Subsumption for reactive control, whereas the other two adopted RAPs. Ullman's (1983) visual routines have also become a foundation for Cognitive Programs (Kruijne and Tsotsos, 2011) and the STAR architecture (Tsotsos and Kruijne, 2014) for general vision. Cognitive Programs are an evolution of visual routines incorporating the full breadth of attentive abilities in vision and enabling an embodiment with eye, head, and viewpoint change.

Belief-Desire-Intention (BDI) (1987). The BDI framework has its origins in Bratman's (1987) theory of human practical reasoning. Practical reasoning refers to thinking about what to do next, as opposed to theoretical or epistemic reasoning, which deals with what to believe in. The traditional desire-belief theory of action maintains that a person performs an action only if they desire a certain outcome and believe that it can be achieved by performing said action. Bratman's contribution was in recognizing that beliefs and desires alone do not cause action but rather lead to intention,

which results in forming a plan and coordinating its execution. According to Wooldridge (1999), intentions play several roles in practical reasoning:

- Intentions drive means-ends reasoning;
- Intentions constrain future deliberation;
- Intentions persist;
- Intentions influence beliefs for future reasoning.

Bratman's conceptual framework was later formalized and implemented by Rao and Georgeff (1991) as an agent architecture. Beliefs were interpreted as the agent's information about the environment and itself, desires were the goals to be achieved, and intentions were commitments to achieving those goals. Beginning with the initial set of beliefs and a set of rules that connect actions to certain conditions, the agents observe the environment and internal events, update their beliefs, respond to new events by forming plans, new goals and intentions, and execute the actions. Then the cycle repeats.

BDI became one of the most popular and studied agent architectures (De Silva et al., 2020). A number of architectures are based on BDI, notably RCS, PRS, dMARS, and JACK. On the epistemic side, Pollock (2000) extended BDI by adding a new component called *likings* and distinguishing different types of desires, which resulted in a new belief-desire-intention-liking (BDIL) model and its implementation in OSCAR.

Curiously, despite originating in philosophy and logic, BDI shares similarities with architectures that have their roots in cognitive psychology. In Soar, for example, beliefs are included in the current state, intentions correspond to selected operators, and desires are goals (Georgeff et al., 1999).

Localist networks. Connectionism is primarily associated with distributed representations, where multiple units (neurons) correspond to a single concept and each unit can in turn be part of representations for many concepts. Another type of network is a localist one where, as the name suggests, representation is more localized; in the extreme case, there is a one-to-one correspondence between the nodes in the network and objects in the world. In general, however, there are many options as nodes technically may correspond to features or any other meaningful entity (Page, 2000). As in any distributed network, inference still takes place by spreading activation from nodes along edges. Localist representations are more transparent and interpretable than distributed representations, and have a similar capacity for generalization and graceful degradation (given enough built-in redundancy).

Localist representations have not been well accepted in the neuroscience community, partly due to fuzziness of the term "localist representation" and concerns regarding their biological validity (although distributed representations and learning methods are also problematic in this regard, as discussed later in Section 11.2). Localist networks are strongly associated with the somewhat pejorative term "grandmother cell," meaning a cell that responds to a specific complex entity, such as one's grandmother (Gross, 2002). Despite ample evidence of explanatory power of localist networks in psychology (Page, 2000) and the presence of highly selective neurons in the brain and artificial networks (Roy, 2012; Bowers, 2017), some go even as far as declaring the localist idea a failure (Barwich, 2019).

In cognitive architectures, localist representations are very common. Perhaps, the most successful example is the ART architecture, which applied

the concept to model many perceptual, attention, and memory phenomena. Another well-known localist architecture is SHRUTI, which investigated reasoning. Clarion, DUAL, LISA, and SPA combine localist and distributed representations to reap the benefits of both. In general, many graph-based representations, such as semantic networks, frames, networks of actions (e.g. RAPs), mentioned earlier, can be construed as a type of localist network and are part of many architectures (see Section 5.2).

2.3 Best of both worlds?

According to Russell and Norvig's (2020) classic textbook, all work in AI is divided along two dimensions: human vs. rational and thought vs. behavior. This produces four possible combinations:

1. *Systems that think like humans.* The goal is to emulate human-like information processing, performance, errors, and biases based on the empirical data provided by introspection, psychological experiments, and neurophysiological studies. As a result, performance, errors, and biases made by an intelligent system should match those observed in humans under similar conditions. Many cognitive architectures fall under this category.
2. *Systems that behave like humans.* Such systems focus on replicating observable human behaviors without the necessary commitment to cognitively and biologically inspired processing (although it happens to a degree). Human performance models are good examples as they are designed to replicate what humans do but often achieve this goal without explicitly modeling human cognitive processes, replacing them with parametric equations or heuristic techniques.
3. *Systems that think rationally.* Such systems are expected to "think right" even if it is not consistent with typical human reasoning in the given the circumstances. Various reasoning and planning engines, and expert systems dominating this category may utilize the same logical inference approaches as humans but generally do not focus on pointing out similarities with biological systems or use them to evaluate their results.
4. *Systems that behave rationally.* Systems that focus on taking the "right actions" without attempting to match human performance. Some examples of systems include autonomous robots and game-playing AI.

In this categorization, the first two groups encompass much of the work on cognitive modeling and cognitive architectures, whereas the last two correspond to what is more commonly perceived as AI. There are, of course, nuances as to what constitutes human inspiration or human-like behavior (see Section 3.2), but regardless of where the line is drawn between human-like and rational, the distribution of works among different categories is highly uneven—nearly 98% of approaches in AI lie within the rational thinking and acting (Sweeney, 2003).

From the AI standpoint, cognitive architectures are a subfield focused on human- and brain-like intelligence, but not all cognitive architecture developers themselves place their work under the umbrella of AI. While Laird (1991) referred to Soar, ICARUS, and AIS as AI architectures, others in the cognitive architecture community acknowledged differences from AI as purely practical

and far removed from human inspiration. For instance, Newell (1981) wrote that AI was fundamentally at odds with psychology, and the gap between them was growing wider. Similarly, Sun (2007) pointed out that despite AI being a constituting discipline of cognitive science in the 1950s, the two have since diverged. AI continued to pursue solutions that were either too domain-specific or not cognitively realistic, while disregarding issues raised by cognitive scientists. Forbus (2010) concurs with this perspective, noting that AI and cognitive science once had quite a bit of overlap and even coordinated their top venues, but this ceased in circa 2001. Besides the lack of interest from AI scientists in modeling human behaviors mentioned by others, he cites financial motivations for pursuing AI and a dismissive attitude toward AI within the cognitive science community as contributing factors.

Regardless of the disagreements on methods and goals, it is hard to deny that both AI and cognitive sciences are rooted in essentially the same problems (Wang, 2006a). Initially, the connection was skewed. Newell (1981) once noted that AI could thrive without psychology but not the other way around—without the tools and algorithms supplied by computer science for analysis and processing of the information, our understanding of how the human mind works could not progress. However, much has changed since the 1980s. Studying the human mind, the only known general-purpose intelligent system, has provided insights into hard problems, for example, by discovering strategies used by biological organisms for solving intractable problems related to intelligence (Van Rooij, 2008; Tsotsos, 2017; Momennejad, 2022).

Even the differences in practicality and cognitive/biological realism between AI and cognitive architectures may not be that significant upon closer inspection. Not all AI research is purely practical, and cognitive architectures do not focus only on theoretical issues. In fact, some cognitive architecture researchers are interested in building models of the human mind that can be useful in practice. Cognitively accurate modeling does not imply inefficiency and may result in smarter systems, although it is entirely possible that more effective non-human intelligence may be discovered eventually (Forbus, 2012). Likewise, cognitive and biological inspiration is not restricted to the domain of cognitive architectures, as there is interest in imbuing AI with more biological details of how humans process information.

Unlike AI, cognitive science seeks mechanistic explanations for mental representations and processes corresponding to observed human behaviors (Brown, 2014). These explanations are expressed in the form of executable programs that can simulate internal processes involved in a given task, and whose final and intermediate outputs can be examined and evaluated against human data. Since the inception of cognitive science as a discipline, thousands of experiments and computational models significantly expanded our understanding of the human mind. However, most of these models are limited to explaining specific phenomena involving a small subset of human cognitive abilities under precisely defined conditions.

Cognitive architectures provide a way of expanding cognitive modeling in scope (the number of phenomena they are able to model), across abstraction levels (theory, algorithms, implementation), and within many domains (tasks and conditions). By design, architectures embody theoretical assumptions in concrete implementations of basic elements of human cognition. Ideally, both the assumptions and algorithmic solutions should be based on the available

psychological and biological evidence, thus bridging the two. Although these properties make architectures arguably more difficult to construct than individual models for specific tasks, this approach could potentially generate explanations of a deeper kind.

What distinguishes cognitive architectures from dominant approaches in both AI and cognitive science is the pursuit of a holistic understanding and modeling of the human mind. Cognitive architectures attempt to provide evidence that particular mechanisms succeed in producing intelligent behavior and thus contribute to cognitive science. Moreover, the body of work represented by the cognitive architectures in this book documents what methods or strategies have been tried previously, how they have been used, what level of success has been achieved, and what lessons have been learned, all important elements that help guide future research efforts. For AI and engineering, documentation of past mechanistic work has obvious importance. But this is just as important for cognitive science, since most experimental work eventually aims to explain how observed human behavior may be generated, for which cognitive architectures provide a rich source of viable ideas and mechanisms.

This puts research on cognitive architectures in a unique position relative to its parent disciplines, allowing it to reap the benefits of both by exploring their ideas while being relatively unaffected by the downsides of either. For example, despite concerns of decreasing interdisciplinarity in cognitive science, cognitive architectures are fairly balanced on this account with about 60% of projects developed by computer scientists, engineers, and roboticists, 35% by psychologists and cognitive scientists, and 5% by neuroscientists.[3]

The influence of AI and computer science is more prominent in cognitive architectures due to the expertise in building software and hardware needed to bring these projects to life. Unlike AI that has seen industry creeping in from the early days, cognitive architectures are predominantly research projects developed in universities where interdisciplinary collaborations are easier to come by and there is less pressure for short-term results. Out of projects discussed in this book, less than a quarter have been developed by government agencies (e.g. NASA, US Army Research Laboratory, US Navy Research Laboratory) and only a few by organizations (OpenCog Foundation, BBN, AOS Group). As such, most cognitive architectures are long-term projects pursuing fundamental ideas rather than quick gains.

Lastly, cognitive architectures are often seen as more theory and research oriented in comparison to AI systems for solving specific problems, such a perception is rather simplistic. Cognitive architectures are hardly a homogeneous group; some aim at answering fundamental questions about human cognition, some are more focused on engineering solutions with what is already known, and some do a bit of both. Among the architectures discussed in this book, the proportions are roughly equal.

2.4 Summary

- AI as a discipline began in the early 1950s with the aim of creating thinking machines. Early efforts focused on areas that were considered difficult for

[3]Based on the main authors' affiliations listed in

most humans (e.g. playing chess). However, deceptively effortless human abilities, such as perception, natural language communication, and commonsense reasoning, quickly emerged as the hardest problems to solve computationally.

- During its history, AI has experienced ups and downs and changes in dominant paradigms. The current revival phase is largely associated with the deep learning methods stemming from connectionism. It is propelled by significant investments from academia and industry and targets practical applications in various domains.
- Born only months apart from AI, cognitive science aimed to explain the mechanisms of human thought and for this purpose drew inspiration and techniques from a wide range of disciplines. Combining experimental and theoretical work with computational modeling techniques greatly expanded our understanding of the human mind and brain. However, there are concerns that the interdisciplinary integration within cognitive science has weakened over the time.
- Research on cognitive architectures is at the intersection of AI and cognitive science. Cognitive architectures pursue a computational approach to modeling human cognition and borrow algorithmic techniques and human experimental data from AI and cognitive science, respectively. Different from AI, the computational approaches used in cognitive architectures are usually grounded in psychology and neuroscience. But unlike cognitive science that studies cognitive phenomena in isolation, cognitive architectures aim to build a holistic model of cognition and interaction between its parts.
- Nearly all cognitive architectures can be traced to a handful of theories and algorithms proposed in the 1960s to 1980s. Means-ends analysis, the blackboard, and graph-based localist representations are among the most popular. As we will see in the next sections, differences in applications, implementation, and various combinations of these elements lead to a large variety of architectures with different theoretical and practical significance.
- Cognitive architectures in many respects represent the best sides of both cognitive science and AI, from which they draw inspiration. Cognitive architectures by design are interdisciplinary and most focus on the fundamental aspects of cognition rather than short-term practical utility.

3 Taxonomies of Cognitive Architectures

In this chapter, we will consider various broad classes of cognitive architectures to better understand properties of these systems and their place within the space of other computational artifacts. Because cognitive architectures are complex systems that combine concepts and methods from multiple disciplines, devising a single classification that fits all systems was nearly impossible, therefore several taxonomies emerged in the literature. We will discuss the two most common categorizations that grouped architectures based on what was perceived as their core properties. In addition, we propose a new classification to categorize architectures based on their proximity to structure and function of the human mind.

Section 3.1 starts with the taxonomy of cognitive architectures by *levels of abstraction* (cognitive, biologically inspired, neural) and embodiment (robotic and simulated) and points out some inconsistencies and overlaps between the categories.
Section 3.2 proposes a new classification based on the *cognitive conformity*, which ranks the architectures based on how well they approximate properties of human cognition. Using this criterion, we define four groups of architectures: inspired, plausible, explanatory, and predictive.
Section 3.3 discusses symbolic, subsymbolic, and hybrid *representations*, as well as their characteristics, strengths, and weaknesses. We describe types of neurosymbolic hybrids, difficulties applying existing taxonomies to cognitive architectures, and propose a simpler categorization based on *level of hybridization*.

3.1 By levels of abstraction and embodiment

In the previous chapters, cognitive architectures were placed within broader efforts toward understanding human cognition and creating AI artifacts. However, besides narrow AI applications and cognitive architectures, there are many other types of intelligent systems. Agency is often seen as the basic requirement that separates intelligent systems from other software and hardware artifacts. Within the set of agent architectures, additional constraints, such as cognitive, biological, and neural realism, and implementation in hardware, further differentiate cognitive, biologically inspired, neural, and robotic architectures. These constraints may apply to some or all aspects of the architecture, such as theoretical basis, implementation, and performance.

A sketch of how different classes of intelligent systems relate to one another in the literature can be seen in Figure 3.1. Agent, cognitive, biologically inspired, and neural architectures form a hierarchy that roughly corresponds to levels of abstraction (see Section 1.1) and

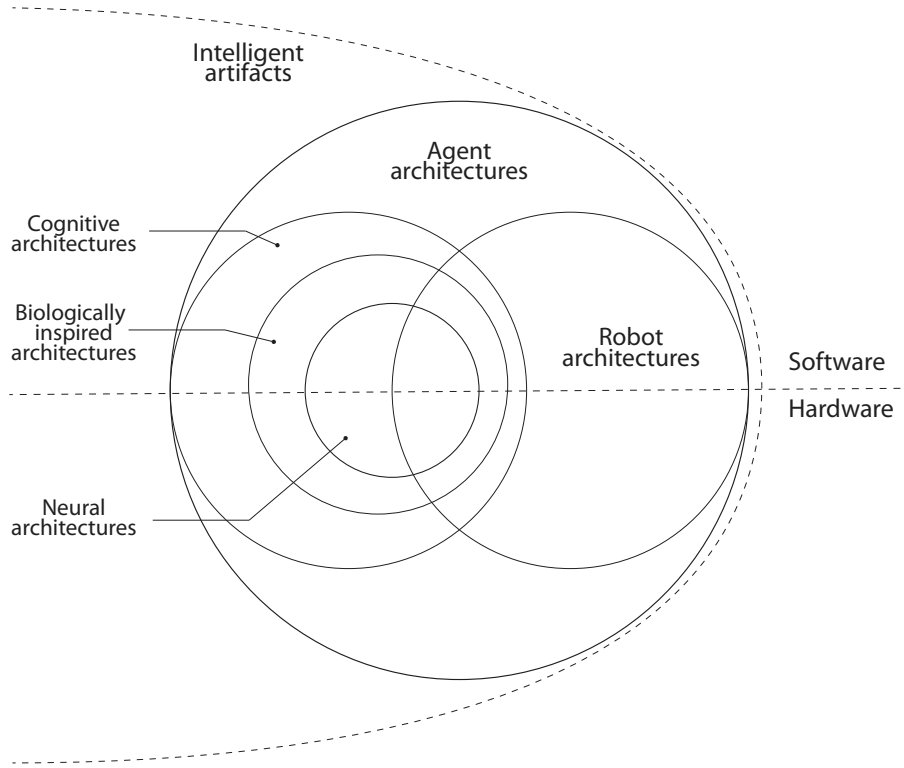

Fig. 3.1 A schematic illustration of the relationships between different types of architectures that form a subset of intelligent artifacts distinguished from other types of software and hardware artifacts (shown with dashed lines). The drawing does not reflect the relative sizes of each subgroup.

Although this classification has become common in the literature, its major drawback is reliance on terms, such as agency and biological plausibility, whose definitions are too broad and interpretations are too varied to draw consistent boundaries between the categories. Next, we will discuss each category in more detail.

Agent architectures

The concept of agency is often described in terms that make it appear synonymous with intelligence. For example, according to a classic definition by Wooldridge (1999), agents are "*computer systems that are capable of autonomous action in some environment in order to meet their design objectives*" which is nearly identical to definitions of intelligence mentioned earlier in Chapter 1). A number of other definitions of agency appear in works of Shneiderman (1995), Franklin and Graesser (1996), Ingham (1997), and Shoham (1999). Using this representative set, we summarize the main features of agency as follows:

- Reactivity—quick response to changing conditions;
- Proactiveness—ability to generate and maintain goal-directed behavior;
- Autonomy—independent operation, which includes choosing to follow another entity's decisions if necessary;

- Social ability—interacting with other biological and artificial entities to achieve goals;
- Temporal continuity—running continuously and maintaining execution state that can be resumed after interruption;
- Adaptation—adjusting to changing environment or task demands;
- Learning—changing behavior based on the previous experience;
- Mobility—transferability to other robotic platforms or systems;
- Personality—individual characteristics, preferences, and emotional state.

Although many of these properties are intuitive, it is a challenge to define them precisely. As a result, the space of possible agents is vast. Wooldridge (1999) noted that even a simple thermostat meets many criteria of agency: it senses (the temperature), has a goal (to maintain temperature), reacts (by raising or lowering the temperature), and works autonomously (without human intervention).

Consequently, agent systems form one of the largest and most diverse subfields of AI. These include autonomous agents, software agents, intelligent agents, interface agents, virtual agents, information agents, and mobile agents in multiple domains including operating systems, satellite imaging data processing, electrical grid management, business process management, air-traffic control, computer games, and more (Luck and d'Inverno, 2003).

Cognitive architectures

Cognitive architectures can be considered a subset of agent architectures because there is a substantial overlap between characteristics of both types of systems (compare desiderata in Section 1.2.2 to requirements of agency above). Beyond agency, cognitive architectures are expected to meet several more criteria, such as cognitive inspiration or plausibility of organization and processing, the ability to explain and predict human cognitive processes and behaviors, and integration of multiple aspects of cognition within a single system. However, interpretations of these properties may vary, making distinctions between agent and cognitive architectures ambiguous and subjective.

Determining the extent to which any given architecture reflects properties of human cognition is not trivial. Let us consider Belief-Desire-Intention (BDI) which is based on Bratman's (1987) theory of practical reasoning and philosophical understanding of rationality. While BDI-based systems are commonly regarded as agent architectures, CoJACK, a BDI system with behavior moderators and additional constraints placed on memory, is referred to as a cognitive architecture (Ritter et al., 2012). Another example is Copycat, a model of analogical reasoning with two key components: a semantic concept network and parallel processing (Hofstadter, 1984). Although neither component is cognitively inspired, the central role of analogy and the parallel, distributed, and non-deterministic operation of the system resembles operations in the human brain. The successor of Copycat, Metacat, extends it by adding metareasoning abilities. Despite sharing theory and implementation, only Metacat is classified as a cognitive architecture, whereas its predecessor is not.

Not all systems categorized as cognitive architectures in the literature predict human behavior. There are, however, specialized human performance models (HPMs) that focus exclusively on this task. Most HPMs are designed to

predict human behaviors and their outcomes in limited domains that involve military and safety-critical civil tasks. Some HPMs are also psychologically inspired and can emulate cognitive processes (e.g. COGNET; Zachary et al., 2000). Given that early research on human performance modeling was aligned with the goals of cognitive architectures (John, 1993), HPMs may be viewed as a type of cognitive architecture. And vice versa, some well-known cognitive architectures, such as, Soar, ACT-R, and EPIC can also be considered HPMs (Kieras et al., 1998; Deutsch and Pew, 2003).

Lastly, there is no predefined list of what aspects of cognition a cognitive architecture should integrate. Arguably important components, such as perception, motor control, and learning, are not addressed in many established cognitive architectures, but it does not disqualify them.

Biologically inspired cognitive architectures (BICAs)

The term BICA originated from a Broad Area Announcement (BAA) issued by the Defense Advanced Research Projects Agency (DARPA, 2005). The agency sought proposals to develop, implement, and evaluate psychologically- and neurologically-based theories, design principles, and architectures of human cognition. Although the BICA program was terminated within a few years of its inception, the term remained widely used.

The purpose of BICAs is to model the human mind-brain more closely than cognitive architectures that are viewed as primarily psychologically inspired. This makes BICAs a subset of cognitive architectures, but the exact boundary between the two is not well-established. Unfortunately, neither the original definition in the BAA nor the subsequent uses of the term provide more clarity. For instance, Samsonovich (2012), in a program paper establishing the BICA Challenge, defines biological inspiration as reproduction of functional properties and internal mechanisms of the human mind, its structures, functions, and dynamics. However, criteria for the BICA Challenge only list targeted human cognitive abilities (which overlap with desiderata Section 1.2.2) without addressing biological inspiration in theory or implementation. Adding to the confusion, projects listed as pursuing BICA are rooted in a broad range of disciplines, including computer science, psychology, cognitive science, and neuroscience. A comparative table of cognitive architectures[1] curated by the BICA society contains an even larger set of architectures and their features, some of which are agent and robot control architectures without clear biological or psychological roots, such as Subsumption and PRS.

In other literature, biological inspiration takes many forms. In some categorizations, representation is the deciding factor. Goertzel et al. (2010b) in their survey contrast primarily symbolic cognitive architectures with biologically inspired ones that have substantial subsymbolic components, distributed representation, hierarchical organization, and excel at pattern-matching and learning.

Others treat biological inspiration as explicit modeling of computational and design aspects of biological systems (Holland et al., 2013). According to this view, BICA aim to be real-life computational equivalents of the human mind in terms of performance, organization, cognitive biases, and embodiment. Such biologically inspired methods may be an intentional attempt to copy

[1] https://bica.ai/architectures/

solutions that exist in nature or emerge from placing the system under the same constraints as biological systems.

A more practical proposal by Love (2021) suggests operationalizing "biological plausibility" as an evaluation against specific data at a specified level of abstraction. This would allow fair evaluation and avoid unnecessary criticisms of high-level cognitive models for not including low-level neural processes and vice versa. For this approach, certain evaluation practices are required, which are yet to be established for cognitive architectures, as we will see later in Chapter 10.

Neural architectures

Neural architectures aim to go a step further than BICAs in emulating the human brain. As a subset of BICAs, these projects are by definition biologically inspired but with additional constraints on the structure of models and acceptable computation approaches that differentiate brain emulation from BICA. Cattell and Parker (2012) list requirements, such as accurate models of spiking neurons, connectivity in the brain, synaptic plasticity, and power consumption. Needless to say, there are very few architectures that attempt detailed models of neural computation. In our selection, only SPA and Leabra qualify.

Robot architectures

Robot architectures (or robot control architectures) have a clear definition: they should be able to control a physical device. In principle, any cognitive architecture can be interfaced with hardware with modifications to account for new sensors and actuators. Architectures that are designed for specific hardware are different: physical properties of the hardware constrain the types of behaviors that the agent can perform and, by extension, the architectural structure and the underlying computational concepts. As a result, these architectures are less general than their software counterparts.

Robot architectures, regardless of their theoretical inspiration, are first and foremost focused on addressing perceptual and motor issues, as well as engineering concerns, including efficiency, reactivity, robustness, power consumption, and graceful degradation in the face of unexpected events. These architectures prioritize good performance on given tasks and implement cognitive abilities to the extent that is needed for achieving this goal.

3.2 By cognitive conformity

In view of the difficulties with the common grouping of architectures into agent, cognitive, biologically inspired, and neural, we propose an alternative scheme. We start by delineating a set of cognitive architectures that include basic modules for perception, memory, and reasoning. Essentially, any system that can accept input, perform inference, and store results of computation fits this description. Within this set, we define four following classes with increasing levels of cognitive and biological detail and additional characteristics that make them more suitable as scientific models of human cognition.

Inspired. Psychologically or biologically inspired architectures use theories, observations, or facts about human cognition to inform the general design of

the architecture or its parts. All that is required is to provide a reference to a theory or theories and verbally or diagrammatically specify what parts are implemented and how.

Plausible. Psychological or biological plausibility requires further proof by referring to concrete theories of the modeled phenomena as well as justification that the implemented model adheres to the theory and produces desired outputs. This can be done by comparing the intermediate and final results of the computation with corresponding human data. In addition to simply producing the desired output, the architecture should have similar temporal constraints, as well as display biases and limitations found in human data.

Explanatory. Ideally, the architecture should explain how internal cognitive processes lead to observed outputs. However, explanatory power is generally more difficult to demonstrate, as most cognitive processes are not directly observable. Therefore, a deeper analysis is needed to tie the computational pipeline to the hypothesized operation of human correlates. One way to do this is to test a range of architectural parameters and analyze the effects they have on the output.

Predictive. The original goal of cognitive architectures should not limited to modeling what is already known about human cognition. Rather, they should aim at deepening our understanding of the human mind and brain by finding gaps in existing theories and proposing new hypotheses. There are two different kinds of testable predictions that cognitive architectures can make:

- Quantitative predictions—output of the model on novel inputs comparable to human data at the appropriate level of abstraction. This can be determined through evaluation on specific tasks and corresponding datasets (see Chapter 10).
- Qualitative predictions—analytically derived and testable predictions regarding previously unobserved patterns and novel explanations of existing phenomena.

The benefit of this taxonomy is that it groups cognitive architectures based on the aspect that matters the most—how well they approximate human cognition. Thus, it takes into account differences between architectures that are vaguely based on folk-psychological concepts and the ones implementing concrete psychological or biological theories that generate testable hypotheses. In addition, gradations in cognitive conformity can be established at different levels of abstraction (e.g. behaviors and neural recordings) and for embodied and disembodied architectures, depending on the types of theory and data used.

3.3 By representation

What abstraction is best for capturing human-level intelligence has been a topic of a long-standing debate that reached its peak in the 1980–1990s. Two camps formed: those defending the symbolic or quasi-linguistic view of cognition, and those who considered connectionism as a more viable alternative. According to the Physical Symbol System Hypothesis (Newell and Simon, 1976), the entirety of mental processing could be described as a computation on atomic symbols. This type of processing is largely inspired by early work

on computer architectures by Turing and von Neumann. A common view of symbolic systems is that they excel at reasoning and planning, but are less capable when it comes to perception and learning in a changing environment. Connectionist systems are primarily associated with the metaphor of the neuron. Rosenblatt's (1958) Perceptron is one of the most known early implementations of this concept that demonstrated its potential for perception and adaptation.

The choice of representation is important, because it predetermines many aspects of the cognitive architecture and in some cases is a principled decision. Not surprisingly, representation-based taxonomies of cognitive architectures are very common in the literature (Vernon et al., 2007; Duch et al., 2008; Goertzel et al., 2010b). In this section, we will discuss in more detail symbolic and subsymbolic representations, their meaning, and practical implications of each paradigm.

3.3.1 Symbolic and subsymbolic

Many accounts point to Newell and Simon (1976) and Feldman and Ballard (1982) as the respective program documents for symbolism and connectionism, respectively. However, both computational approaches were formulated decades earlier; Craik (1943) was one of the first to express the view that the human brain operates with symbols, whereas the term connectionism was introduced by Hebb in the 1940s according to Elman et al. (1996).

Symbols and neurons

Symbolic systems, as the name suggests, operate on symbols defined as physical tokens (e.g. strings or variables in a programming language) that refer to some concept. Symbols are atomic in the sense that they cannot be broken down into parts and cannot change their form. Some symbols may also refer to compound concepts or expressions. For example, the concept of "bachelor" combines two other concepts—"unmarried" and "man." This still does not violate atomicity as neither word is a constituent part of "bachelor," which itself is treated in expressions as a singular entity (Chalmers et al., 1992).

Connectionist systems are composed of many identical and simple computational units that are connected to one another and together form a graph structure (network). Both nodes and edges in this graph are assigned weights (activations) that dynamically change depending on the input. Concepts are represented not by the individual nodes, but rather by patterns of node activations within the network. Therefore, each node can be part of multiple patterns of activations and contribute to representations of multiple concepts.

Syntax and semantics

Symbolic and subsymbolic representations differ in composition and meaning that they convey about concepts. Symbols and relationships between them are combined following syntactic rules. However, these syntactic manipulations operate on the shape of the tokens, not their meaning. Thus, the label of the symbol can be arbitrary and replacing any symbol with any other does not change the relationships between the elements within the given expression. Syntactically, "Mary gave a book to John" is equivalent to "X gave Z to Y," both of which express a relationship between two symbols that refer to abstract

or real objects. Semantic meaning is provided externally by whoever defined and used the expression. As a result, neither the symbol nor the syntactic manipulation is capable of relating symbols to objects in the real world (without interpretation assigned externally). This phenomenon is referred to as the symbol grounding problem, initially triggered by the Searle's (1980) Chinese Room argument and given its classical formulation by Harnad (1990).

Connectionist systems operate on nodes and edges that can be interpreted as concepts but themselves are one level below symbolic, hence the name subsymbolic. Although individual nodes do not carry semantic meaning, the distributed patterns of activity over the nodes do. Unlike symbolic architectures where semantic meaning is supplied externally by the programmer, connectionist systems obtain it directly from the environment. However, the concepts learned this way might not be readily available for analysis and interpretation because the learning procedure for individual or composite concepts does not preserve the integrity of its components. For example, a self-supervised neural network can learn how to distinguish between different classes of images and form representations for both that are neither linguistic nor easily decipherable.

That is not to say that connectionist systems are immune from the symbol grounding problem. Although some categories can be derived from sensory experience through correlation alone as Harnad (1990) and others suggest, this process leaves out the origins of categories for non-existent things and ignores evidence for top-down formation of categories (Christiansen and Chater, 1993).

A common solution is grounding internal representations through task and interaction, which can work with both symbols and networks. One demonstration of how it may occur is a guessing game study by Steels et al. (2005) with two agents—a speaker and a hearer—that communicate to reach a cooperative goal. The speaker asks the hearer to point to an object of a certain color, and the hearer tries to understand and execute the request. Both agents initially have no context, therefore the one assigned the role of a speaker initially comes up with random names for previously unseen categories of colors. As a hearer attempts to perform the requests, the shared lexicon of both agents expands, and the meaning of the individual terms becomes more precisely defined.

Strengths and weaknesses

Despite differences in processing, syntax, and semantics, both symbolic and subsymbolic representations are computationally equivalent to a Turing machine, can be implemented in the same programming language, and run on the same von Neumann computer architecture (Arbib, 1961; Newell and Simon, 1976). But even though it is theoretically possible to express a connectionist network as a symbolic system consisting of rules and vice versa, it is often not feasible in practice. Therefore, the choice of representation is a trade-off.

Pattern recognition capacities in perception, motor control, categorization, and associative memory eluded implementation in symbolic formalisms (Dinsmore, 1992). Thus, virtually all practical systems that must deal with the high-dimensional perceptual data use connectionist methods. At the same time, neural networks historically struggled with tasks that required symbol manipulation, such as planning and logical inference. Even today's large vision and language models still cannot reliably solve planning and reasoning problems (Valmeekam et al., 2023; Rahmanzadehgervi et al., 2024).

Symbolic systems learn mainly by rote memorization, in other words, they need an explicit knowledge base that specifies entities and relationships between them for a particular domain or task. While it is often considered a weakness of symbolism, sometimes the ability to simply tell the system what to do can be useful. For neural networks, direct upload of information is not an option. Instead, learning even a simple classification task requires a large number of training instances, and even more data is needed if the system has to solve a task without supervision. Once knowledge is acquired, further learning is more difficult with both types of representations but in different ways. Symbolic systems struggle to form new associations and often break when input is noisy or unexpected. Connectionist systems are less brittle, but their flexibility can easily backfire. One such problem is catastrophic forgetting, when new information causes irreparable damage to what was learned earlier (French, 1999).

Connectionist systems capture some features of neural computations, such as distributed representation and parallel processing. However, these similarities are fairly superficial, as most neural networks do not represent the biological neurons functionally or anatomically (see Section 11.2). Like brains, neural networks resist direct examination and thus cannot be easily interpreted. Symbolic systems, on the other hand, offer transparency but lack any connection to neural processes.

3.3.2 Neurosymbolic integration

How can apparent differences between symbolic and subsymbolic representations be resolved? The simplest, albeit radical, solution is to ignore one of the representations entirely and focus on the other. This has been tried before with symbolism in the 1980s, but without success, as we saw earlier in Section 2.1. At the time of writing this chapter, another attempt is unfolding, this time with connectionism, and some troubling trends are already becoming apparent (see Section 11.2).

A more moderate option is to consider that connectionism and symbolism are Turing equivalent and thus can perform the same computations. The choice then becomes a matter of preference or suitability for the given purpose. In fact, this is the route taken by the majority of cognitive architectures that combine elements from both paradigms.

Types of hybrids

There are many ways of modeling cognition using a mix of symbolic and subsymbolic representations, and several taxonomies have been proposed to categorize them. The most detailed are taxonomies by Hilario (1997) and Bader and Hitzler (2005).

The scheme proposed by Hilario (1997) and illustrated in Figure 3.2 is very comprehensive and captures nearly all possible hybridization options. The top-level division is made between the unified (purely connectionist) and hybrid (mixed symbolic/subsymbolic) approaches. The unified strategies are further subdivided into neuronal (biological) and connectionist (non-biological) approaches, the latter with localist, distributed, or mixed representations. Hybrid strategies are comprised of translational or functional hybrids. Functional hybrids are composed of separate symbolic or connectionist modules and are

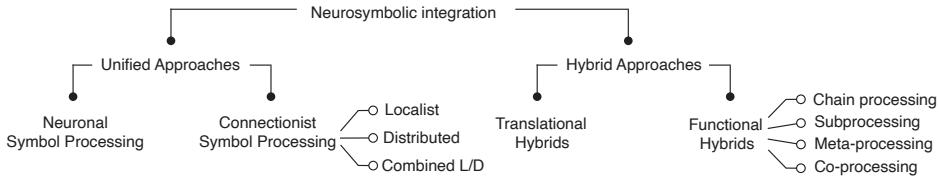

Fig. 3.2 Taxonomy of possible combinations of symbolic and subsymbolic representations proposed by Hilario (1997). Top-level division is made between unified connectionist systems that give rise to or operate with symbols and hybrid approaches that combine symbolic and subsymbolic processing in various ways.

further subdivided based on how tightly the modules are integrated and what part of the processing pipeline they occupy. Translational hybrids incorporate only partial symbolic and connectionist representations, such as symbols processed by neural networks, or symbolic patterns extracted from neural networks. Therefore, translational hybrids are an intermediate stage from unified to functional hybrids.

Bader and Hitzler (2005) proposed to arrange hybrids along three dimensions: interrelation, language, and usage, each with additional subdivisions. Unlike the hierarchical taxonomies, the dimensional approach better captures the fact that certain features of hybrids are independent. The first dimension, interrelation, largely follows Hilario's taxonomy and contains the following groups: integrated-hybrid, neuronal-connectionist, local-distributed, and standard-nonstandard. The last distinction applies to standard neural networks that learn via backpropagation and custom ones that have different units and learning methods (e.g. SHRUTI). The language dimension describes whether the symbolic part of the system uses symbolic or logical languages. Finally, the usage dimension is somewhat similar to translational hybrids in Hilario's taxonomy and has two subgroups: extraction-representation and learning-reasoning. The first reflects whether symbols are extracted from the connectionist network or represented within it. The second determines whether learning or reasoning is prioritized.

Other taxonomies cover some portions of these two categorizations. For example, one of the earliest schemes by Medsker and Bailey (1992) distinguishes between standalone-symbolic, standalone-connectionist, and transformational models. Within the latter category, symbolic representation is converted to subsymbolic via loose, tight, or fully integrated components. Transformational systems in this taxonomy combine translational and functional hybrids in Hilario's taxonomy. Ultsch (1998) uses the term cooperative to describe modular hybrid systems where no transition between symbolic and subsymbolic knowledge takes place, which only happens in true hybrids systems (translational hybrids in Hilario's taxonomy). Another well-known categorization by Sun and Wermter (2000) is an abridged version of Hilario's. It distinguishes between unified localist and distributed connectionist systems, transformation architectures and modular hybrids with loosely, tightly, or fully integrated components. Two directions of hybridization are suggested: vertical, along with levels of computation, and horizontal, where multiple computational levels are mixed for practical considerations. The most recent taxonomy of neurosymbolic AI by Kautz (2022) considers six types of hybrids which also can be mapped onto Hilario's (1997) classification.

Applying taxonomies

Although taxonomies described above are detailed and consider many possible combinations of symbolic and subsymbolic components, applying them in practice to categorize arbitrary cognitive architectures is often difficult.

Identifying representations. Even the first step in categorizing hybridization type—identifying symbolic and subsymbolic elements—is not trivial, because there is little consistency in how representations are defined in the literature. Furthermore, out of the architectures we reviewed for this book, two-thirds omitted the specification of the representation and only a few reported on the particular symbolic/subsymbolic processes or elements.

It is universally agreed that strings of characters, frames, production rules, and non-probabilistic logical statements are symbolic and distributed representations, such as neural networks, are subsymbolic. However, there are a number of representations whose status is less certain. For example, probabilistic action selection is considered as symbolic in CARACaS (Huntsberger and Stoica, 2010), CHREST (Schiller and Gobet, 2012), and CogPrime (Goertzel, 2012b), but is described as subsymbolic in ACT-R (Lebiere et al., 2013) and Copycat/Metacat (Marshall, 2006). Numeric data is treated as symbolic in CAPS (Just and Varma, 2007), BB1 (Hayes-Roth, 1995), and EPIC (Kieras and Meyer, 1996), but is regarded as subsymbolic in SASE (Weng, 2002). Similarly, there are disagreements regarding the status of the activations (weights or probabilities) assigned to symbols, use of reinforcement learning and image data, etc. For example, reinforcement learning is considered symbolic in CARACaS (Huntsberger and Stoica, 2010) and CHREST (Schiller and Gobet, 2012), and subsymbolic in MAMID (Reisenzein et al., 2013). Images are considered subsymbolic in MAMID (Reisenzein et al., 2013) and SASE (Weng, 2002), and non-symbolic (or iconic) in Soar (Laird, 2012b) and RCS (Albus and Barbera, 2005).

Identifying hybridizations. Mismatch between the level of detail in the description of the architecture and description of class can also cause issues. More detailed and nuanced distinctions are difficult to identify in the literature. Conversely, if the category is not precisely specified, it can apply to too many systems. Even if descriptions of the categories and cognitive architectures are complete, interpretations may differ. For example, according to Hilario's (1997) classification, modules in the blackboard architecture are loosely connected, but the authors of blackboard architecture consider the coupling tight since there are both bottom-up and top-down flows of information (Hayes-Roth et al., 1989).

Gaps in taxonomies. Hybridization within the architecture may happen in many places at once: an architecture can be interfaced with another architecture with different representation and also contain a mixture of representations within its modules. Such scenarios cannot be captured with the tree-like taxonomy, unless duplicate leaves are permitted. Placing the architectures in a continuous multidimensional space spanned by several axes resolves this issue, but becomes difficult to visualize and comprehend. Another drawback of common taxonomies of neurosymbolic integration is that they do not permit representations that are subsymbolic but are not necessarily connectionist networks, such as various features used in computer vision, e.g. SIFT (Lowe, 2004) and other interest point detectors.

A modified taxonomy

Because many fine-grained types of neurosymbolic integration cannot be easily discerned from the descriptions of architectures, we propose a simpler taxonomy based on the levels of hybridization.

Intra-modular hybrids are either unified (i.e. composed of a single module) or modular systems where a hybrid representation is dominant. A common example is the implementation of memory via semantic networks that combine symbolic and subsymbolic elements. For example, ACT-R, Soar, CAPS, Copycat/Metacat, CHREST, Clarion, and NARS combine symbolic concepts and rules with subsymbolic elements, such as activation values, spreading activation, stochastic selection process, reinforcement learning, etc. Some architectures take this concept to the extreme. For example, DUAL consists of a large number of highly interconnected hybrid agents, each of which has a symbolic and subsymbolic component integrated at a micro-level.

Inter-modular systems consist of multiple separate modules or processes, some of which are symbolic, subsymbolic, or hybrid. This is particularly common in robotic architectures. For example, in 3T, ATLANTIS, RCS, DIARC, and CARACaS, a symbolic planning module determines the behavior of the system and one or more modules are used to process visual and audio sensory data using subsymbolic techniques like neural networks and optical flow calculation. Cognitive architectures, such as STAR and ARCADIA also combine symbolic knowledge base and reasoning with subsymbolic image processing algorithms. A typical inter-modular architecture is implemented as a set of interconnected competing and cooperating modules, where individual modules are not restricted to a particular representation (Kismet, LIDA, ISAC, CORTEX, Polyscheme, FORR).

Super-modular hybrids either encapsulate another architecture or combine multiple complete or partial architectures. Even though additional effort is required to build interfaces for communication between them, this approach takes advantage of the strengths of each architecture. Naturally, this approach is most beneficial when two (or more) architectures in question are different, for example, use different representations and processing styles.

Many of such hybrids are proof-of-concept one-off experiments, to demonstrate the feasibility and utility for certain tasks. Some examples include CERA-CRANIUM/Pogamut (Arrabales et al., 2009b) and CERA-CRANIUM/Soar (Asensio et al., 2014) hybrids for playing video games and IMPRINT/ACT-R for human error modeling (Lebiere et al., 2002). A good overview of conceptual and technical challenges involved in creating an interface between cognitive and robotic architectures is given by Scheutz et al. (2013), who develop and test their integration framework on two pairs of architectures: ACT-R/DIARC and ICARUS/DIARC.

We are aware of only a few long-term super-modular projects, among them SAL and ADAPT. SAL combines ACT-R and Lebra such that ACT-R is used to guide the learning of Leabra models (Jilk et al., 2008). In the robotic architecture ADAPT, Soar is utilized for control, whereas separate modules are responsible for modeling a 3D world from sensor information and for visual processing (Benjamin et al., 2013).

Overall, hybrid architectures are the most numerous and diverse group (see Figure 3.3) that forms a continuum between connectionist and symbolic

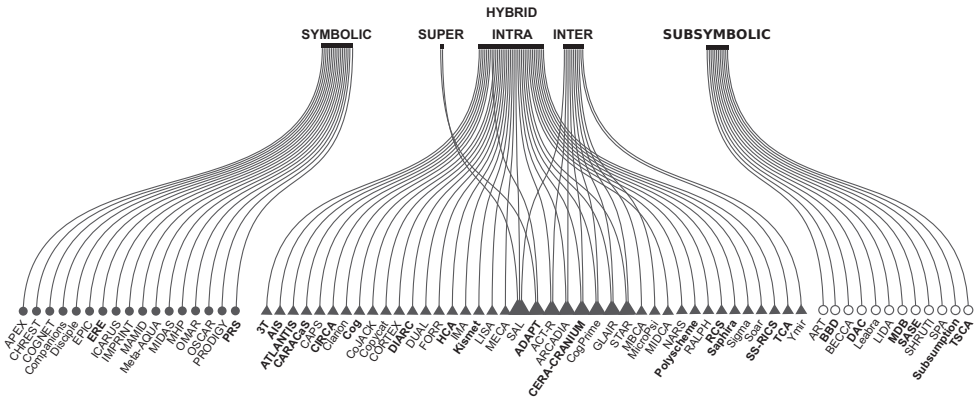

Fig. 3.3 Taxonomy of cognitive architectures according to the new symbolic-subsymbolic hybridization scheme. Symbolic, hybrid, and subsymbolic architectures are shown with filled circles, triangles, and empty circle symbols, respectively. Hybrid cognitive architectures are further split into super-, intra-, and inter-modular as defined in the text (larger triangles indicate that more than one hybridization strategy is present). Architectures shown in bold were implemented on a robotic platform. Sorting order is alphabetical within the subgroups.

systems, depending on the proportions and roles played by symbolic and subsymbolic components. Although quantitative analysis of this space is not feasible, it is possible to crudely subdivide it. For instance, some architectures such as CogPrime and Sigma are conceptually closer to connectionist systems, as they share many properties with the neural networks. On the other hand, CHREST and RALPH, as well as the architectures implementing symbolic subprocessing, 3T and ATLANTIS, are primarily symbolic but utilize subsymbolic elements, such as probabilistic reasoning and learning mechanisms.

3.4 Summary

- Taxonomies of cognitive architectures discussed in this chapter focus on three aspects of cognitive architectures: levels of abstraction and embodiment, ability to model and explain human cognition, and representations (symbolic, subsymbolic, and hybrid). These broad categorizations help organize the extensive literature on cognitive architectures and related phenomena.
- A defining feature of architectures that distinguishes them from other software and hardware artifacts is agency, understood broadly as the ability to deliberately act in a complex environment.
- Common categorization of agent architectures represents them as a nested hierarchy of cognitive, biologically inspired, and neural architectures that are designed to ensure psychological, biological, and neural realism, respectively. Robot architectures can belong to either group and are defined by the ability of the architecture to control a physical device.
- The categories of cognitive, biologically inspired, and neural architectures are not well defined and can be conflated with levels of abstraction and representation. We suggest an alternative categorization *by cognitive conformity* that focuses on how well the architecture models human cognition with

four classes: 1) architectures inspired by findings in psychology, cognitive science, or neuroscience, 2) those implementing plausible mechanisms, 3) architectures that can explain cognitive phenomena beyond matching human input-output data, and 4) architectures that can propose novel testable hypotheses.

- Another common taxonomy is by internal representations: symbolic, subsymbolic, and hybrid combining both. Internal representations define many properties of the intelligent system, such as their general organization and applicable algorithmic solutions. Because symbolic and subsymbolic approaches are complementary, the majority of the architectures incorporate both to exploit their benefits and mitigate limitations.
- Existing neurosymbolic taxonomies describe multiple ways of combining symbolic and subsybmolic representations. In practice, however, these categorizations are difficult to apply to cognitive architectures due to the large variety of approaches, nuanced differences between them, and insufficient technical details to identify relevant elements. Therefore, we outline a simplified categorization based on the three *levels of hybridization*: intra-, inter-, and super-modular.

Part II

HOW ARE COGNITIVE ARCHITECTURES BUILT?

Part II overviews core components present in most cognitive architectures and their associated human cognitive abilities. Here, we provide many concrete examples of theories and implementations and, whenever possible, point out their strengths and weaknesses with respect to relevant human competencies.

The chapters are organized in the order of processing in a typical cognitive cycle, starting with sensation and perception (Chapter 4) that help extract information from the environment. We then investigate the role of memory in storing and managing knowledge (Chapter 5), as well as learning new knowledge (Chapter 6). Next, we look at how obtained knowledge affects action selection and reasoning (Chapter 7). Lastly, Chapter 8 examines how these different components are organized to form a single processing pipeline with various information flows within it.

4 Sensation and Perception

Any intelligent system, whether biological or artificial, uses sensors to perceive its surroundings. For most of us, perception is immediate, effortless, and highly accurate, but in reality it is an incredibly complex and still not fully understood process.

In this section, we will look at what sensory information cognitive architectures obtain from the environment, how they process it, and how their perceptual abilities compare to those of humans in terms of quantity and quality of available sensory modalities, fidelity of sensors, and transformation of raw sensory data into representations useful for carrying out tasks. This chapter is structured as follows:

Section 4.1 provides an overview of human sensory modalities and corresponding modalities (and associated sensors) in cognitive architectures.
Section 4.2 focuses on the properties of real and simulated environments relevant for perception.
Section 4.3 is dedicated to vision as one of the most important and well-represented sensory modalities. Here, we introduce visual processing stages and describe implementations of these stages with physical or virtual sensors in different types of architectures.
Sections 4.4–4.7 discuss other sensory modalities, such as somatosensation, audition, nociception, olfaction, and gustation.
Section 4.8 covers multimodal perception, which modalities are frequently combined, and the advantages of doing so.
Section 4.9 describes visual attention, types of attentional mechanisms, their properties, and implementations.

4.1 Sensory modalities

Because senses are our only connection to the outside world, our perception of reality is shaped by the quality and quantity of the sensory information we receive. The same holds for any artificial system—quantity, quality, and diversity of available sensors determine what it can experience and respond to. In this section we will discuss what humans can perceive through their senses and how this sensory experience can be approximated computationally with physical or virtual devices.

4.1.1 Human senses

In 350 BC, Aristotle introduced the earliest known taxonomy of human senses (Hicks, 1907, p. 109). It listed five distinct senses based on corresponding visible sensory organs—sight (eye), hearing

Table 4.1 A list of human sensory modalities and corresponding physical sensors used in cognitive architectures. Pain is the only sense that has no associated sensors because it is simulated in all cognitive architectures that have it.

Sense	Physical sensor
Vision (seeing)	Cameras, range sensors
Audition (hearing)	Microphone
Gustation (taste)	pH, humidity, and conductivity sensors
Olfaction (smell)	Chemosensors
Haptics (touch)	Force, touch, and pressure sensors
Vestibular sense (balance)	Gyroscope, accelerometer
Somatosensation (proprioception, pain)	Actuator feedback, odometer, IMU

and touch (skin). This categorization persisted for over two millennia and is still in use today.

Only during the last century were the number of senses, criteria for differentiation between them, and independence of processing within and across modalities investigated more systematically. Currently at least seven sensory modalities are recognized: the Aristotelian five and two more—vestibular sense (balance) and somatosensory modalities (proprioception, pain) (Kandel et al., 2021). These modalities are determined based on the physical stimulus, receptor type, neural path, and cortical receiving area (Mather, 2016). Hearing, vision, touch, and balance have distinct pathways and physical stimuli (sound, light, mechanical and motive force). Smell and taste share stimulus (chemical contact) but have distinct neural pathways. Somatosensation shares cortical destination with touch, but has separate receptors and pathways.

Even though modern categorizations are more principled than Aristotelian taxonomy and grounded in current findings from biology and neuroscience, they still present an incomplete view of human sensory perception. There are potentially many other senses, such as body temperature, hunger, thirst, suffocation, fatigue, and many others, that are not supported by the standard taxonomies (Macpherson, 2011; Ritter and Serdiuk, 2024). Furthermore, treating senses as independent is also difficult to reconcile with discoveries of pervasive multisensory integration and phenomena such as synesthesia (e.g. seeing colors when listening to sounds) and sensory substitution (e.g. using touch to compensate for impaired vision) (Fulkerson, 2014). However, since these topics have not yet been investigated in the context of cognitive architectures, we will focus only the basic sensory modalities and consider each separately.

4.1.2 Sensory modalities in cognitive architectures

We begin by examining sensory modalities supported by cognitive architectures and corresponding artificial sensors. To do so, we tallied up all mentions of sensors in publications for each architecture in our selection, noting which sensors were physical (real) or virtual (simulated) devices. Each type of sensor was associated with one of the seven sensory modalities defined by Mather (2016), as summarized in Table 4.1. Note that some cognitive architectures have been implemented on a variety of robotic platforms with different sensor suites and as a result are associated with dozens of sensors. Thus, to avoid

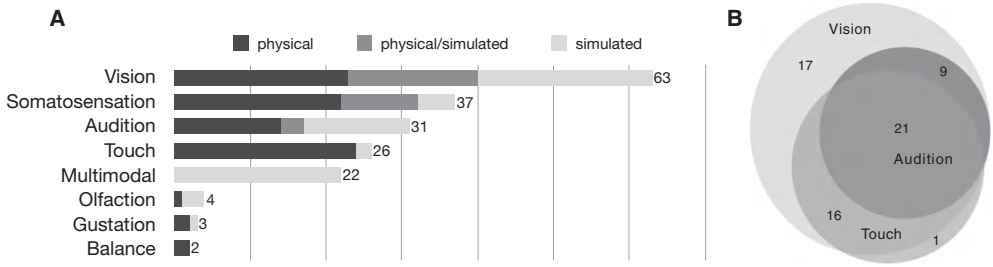

Fig. 4.1 A summary of sensory modalities in cognitive architectures. A) Each bar represents the number of cognitive architectures implementing a respective modality using physical sensors, simulated sensors, or either. B) A Venn diagram showing the number of architectures implementing combinations of one, two, or three of most common sensory modalities: vision, audition, and somatosensation.

bias, we determined for each architecture only whether a particular sensory modality was implemented in some capacity (physical, virtual, or both), regardless of the type and number of sensors used. To determine the potential for multimodal perception, we also noted all cases when two distinct modalities were present in a single implementation.

The resulting diagrams in Figure 4.1 show distributions of sensory modalities and sensor types in our sample of cognitive architectures. Overall, the following general observations can be made based on the gathered data:

- Even with our generous counting criteria, more than one-third of cognitive architectures implement a single sensory modality and only few implement more than three.
- Vision is by far the most common sense, closely followed by somatosensation and audition, with the remaining senses far behind. Only a handful of cognitive architectures have olfaction and gustation.
- Virtual (simulated) sensors are quite common, particularly for visual, proprioceptive, and auditory modalities. In other words, many architectures do not implement any sensory processing and rely instead on the symbolic or numeric representation of the world provided by the simulated environment.
- Although not shown in the diagram, a closer examination of the data reveals that sensory abilities of different cognitive architectures correlate with their type and knowledge representation. Specifically, we found that physical sensors were more prevalent in the hybrid and robotic architectures, whereas symbolic architectures were more likely to rely on simulated sensors.

4.2 Environment

Sensory abilities of living organisms evolve not only to satisfy their goals, but also to fit the properties and demands of their habitat (Oteiza and Baldwin, 2021). Cognitive architectures are likewise linked with the environment and task, but through intentional design rather than gradual adaptation. Consequently, environments used for developing cognitive architectures should be considered carefully; it may be tempting to simplify settings for ease of development and experimentation, however the resulting systems may not accurately represent the intricacies of real-world perception. Thus, we will

begin by considering the real and simulated environments used for research on cognitive architectures.

Real environments. Uncontrolled real-world environments are the most demanding because of their scale, unconstrained layout, potential of exposure to elements, varying lighting conditions, and higher chance of unexpected events occurring. As a result, only a few cognitive architectures are able to operate in such conditions. One example is driving on a public highway with traffic (Murphy et al., 1994) and off-road test track at 35 km/h (Coombs et al., 2000) in an autonomous vehicle controlled by 4D/RCS. Another real-world test of note was conducted in an industrial park with steep hills and grassy soft ground with the TCA architecture on an Ambler hexapod robot (Krotkov and Simmons, 1996). Lastly, a Mars-like environment strewn with rocks and boulders was used to test capabilities of ATLANTIS aboard a planetary rover (Gat, 1991a).

Controlled real environments offer known and fixed ambient conditions and layout. Typically, these are indoor locations, such as a lab or an office, hallways, or even a tabletop. Such settings are useful for tasks involving navigation, interaction with other agents, assistive applications, and object manipulation.

Simulated environments. Simulations have been a staple of research in artificial intelligence (AI) since its inception, and for a good reason. Compared to the real world, simulations provide full control over the environment, recreate unsafe or difficult to achieve conditions, and can run at artificially high speeds for many iterations without the risks of injuring humans or breaking the hardware. It should come as no surprise that only about one-third of the architectures in our selection have been implemented on a hardware platform and thus can run in real environments. The rest operate in simulated environments with various degrees of fidelity.

Perhaps, the most sophisticated simulations are digital copies of existing environments that allow experimentation with high-quality real-world data. An early example of this approach has been demonstrated on a rover controlled by the ATLANTIS architecture. Recordings of the challenging outdoor terrain imitating conditions on Mars were used as a world model in simulated experiments (Gat, 1991a). Although such detailed simulations are useful for bridging real and simulated environments, they are laborious to record and maintain and are not common in the field of cognitive architecture research. In modern research on embodied AI, however, virtual replicas of real environments are being used extensively (Duan et al., 2022) and may be a good option to consider in the future.

A step below are 3D simulations combined with physics engines and agent models. These simulations can be quite realistic, but do not necessarily represent any existing environment in full detail. Gazebo is one of the commonly used open-source simulations of this kind, which provides access to physics engines, photorealistic rendering of the scenes with lighting, shadows, and textures, and accurate sensor data with noise models from various sensor types (Koenig and Howard, 2004). Gazebo has been used to create various environments for testing cognitive architectures: tabletop setting with common objects for DIARC (Wilson et al., 2016), town street with buildings and roads for LIDA agent (Madl et al., 2016), and indoor environment with a model of Baxter humanoid robot for MIDCA (Cox et al., 2016).

The simplest kinds of simulations are discretized 2D worlds that represent the environment at different levels of granularity. Placement and movement of objects in the environment usually occurs according to predefined rules. What can be perceived about the world is also defined rather rigidly, usually as a set of predicates provided automatically or when requested by the agent.

While development in simulation is easier initially, the architecture will likely require considerable changes to account for added environment complexity and inevitable sensor noise present in the real-world conditions. Even though many simulations seem "acceptable" to human eyes, they are still abstractions controlled by their designers and results obtained through such means may be misleading. In robotics, closing the gap between simulation and reality (sim-to-real) still poses a considerable challenge, despite progress in generating realistic imagery and physics (Zhao et al., 2020).

Sim-to-real transfer is even more acute for the simplified simulated environments that the majority of cognitive architectures use. As a result, few architectures developed in simulation have been tested in the real world, and, even then, under highly controlled conditions. For example, ACT-R, Soar, and DUAL have been eventually interfaced with robots but as isolated proof-of-concept experiments rather than a permanent move away from simulation. On the other hand, specialized robot control architectures initially designed with physical bodies in mind can be deployed on different robotic platforms or in simulated environments. In fact, many use simulation as a convenient way of transferring skills to inaccessible domains or to avoid damage to equipment during experimentation. Some successful examples are RCS and ATLANTIS, both tested on a variety of physical platforms and simulations.

4.3 Vision

Historically, vision has been considered a dominant human sensory modality (Posner et al., 1976). While recent works suggest a more integrated view of human sensory experience (Stokes and Biggs, 2014), cognitive architectures remain vision-centric; if any given architecture has only one sense, it is almost certainly vision. Given the prevalence of vision in cognitive architectures and significant effort spent on this sensory modality compared to others, it deserves a more detailed consideration. In what follows we will discuss sensors, physical and virtual, and their properties with respect to human visual capabilities, stages of human visual processing, and their implementations in cognitive architectures, as well as common techniques for optimizing visual processing.

4.3.1 Sensors

Various cameras are the most common sensors used for visual modality. However, because depth is difficult to extract from images, range sensors are often used in addition to or even instead of cameras, depending on the tasks. Here, we will discuss how well these types of sensors approximate human vision.

Cameras

Cameras by design mimic some optical properties and functions of the human eye, which makes them a natural choice of vision sensor. A camera has an

opaque body with an aperture which is similar to the sclera and pupil of the eye. A system of lenses in the camera refracts the light, although it typically lacks the flexibility of the eye's cornea and lens. A digital sensor converting photons into electric signals is similar to the retina, a light-sensitive layer of tissue covering the back of the eye.

Despite some common design elements, there are significant differences between the camera and the biological eye. Unlike digital sensors and film, the eye's retina is not uniformly sensitive to light because of the uneven distribution of receptors. Because most of them are located in the central region (fovea), visual acuity (resolution) is highest there and drops off quickly toward the periphery. As a result, unlike a camera that passively processes the entirety of the scene it is pointed at, human vision is an active process that involves eye movements to bring different areas of the scene into the fovea, continuously changing the contents of the retina.

Although human vision is not the best in the animal kingdom and the optical quality of the human eye is low compared to some artificial systems, the human optical system is highly optimized (Navarro, 2009; Artal, 2015). In fact, in the process of evolution, human eyes reached an absolute terminal point that closely approximates limits imposed by physics of the eye (Rose, 1973). Thus, any modifications to the human eyes, whether intentional or due to aging, usually degrade their performance.

The extent to which technology can replicate human vision is unknown, as there are not many direct comparisons between the characteristics of available cameras and human eyes. The only comprehensive study by Skorka and Joseph (2011) is now over a decade old, but gives some idea. The authors benchmarked twenty-four commercial and research digital cameras against an idealized human eye with respect to eight parameters measuring geometric properties (visual field, spatial/temporal resolution), signal and noise power, and power consumption. The conclusion was that none of the cameras rivaled the human eye in any of these characteristics, with particularly wide differences in dynamic range. This is not surprising, as human eyes are remarkable at dark adaptation, being capable of increasing their sensitivity a thousandfold in a short span of time (Rose, 1973).

Results of the Skorka and Joseph's (2011) benchmark hold for cameras used in implementations of cognitive architectures we considered. Even the recent architectures do not make use of the most advanced sensors available today. As a result, there are large discrepancies between characteristics of the sensors and corresponding properties of human vision, which will be discussed next.

Field of view (FOV). Humans have a wide-angle visual field, spanning 200° horizontally and 125° vertically (Strasburger, 2020). In comparison, cognitive architectures used a wide variety of cameras. Most were narrow angle, such as 35° (BBD; Seth et al., 2004b), 45° (ADAPT; Benjamin et al., 2006), or 58° (3T; Wong et al., 1995). In some cases wider angle cameras (e.g. 115° FOV camera used in RCS; Camus et al., 1996) or multicamera setups providing 360° panoramic view (CARACaS; Wolf et al., 2010; Saphira, Konolige et al., 1999).

Binocular vision. The human visual system relies on two eyes, but not only as insurance against loss of an important perceptual organ. Approximately 120° of the human visual field is binocular, which offers many advantages, most notably, stereopsis (ability to discern depth of static and moving objects) and

binocular summation that enhances vision quality. Many cognitive architectures support stereo vision, particularly for tasks, such as navigation, object manipulation, and social interactions where depth is necessary for adequate performance.

Acuity/resolution. The human retina packs 127 million photoreceptors in an area of 1100 mm^2 most of which are concentrated in the center, which has a resolution of 1/60th of a degree, sufficient to discern very fine details, such as distinguishing HD video from 4K. Ten years ago, state-of-the-art digital sensors featured 16.6 million photo receptors evenly distributed over 1,600 mm^2 sensor area (Taylor, 2011). Today, technology can fit 200-megapixels into a 75 mm^2 sensor (Omnivision, 2022). However, most architectures use inexpensive consumer cameras that produce low-resolution images rarely exceeding 1,000 pixels in either dimension and are often downsampled further to speed up processing. For example, Darwin VII (an instance of BBD) navigates the environment using a single camera with resolution of 230×240 px, downsampled 4x (Seth et al., 2004a). Likewise, 3T processes 1,024×1,024 px images downsampled to 793×793 px (Wong et al., 1995).

Peripheral/central vision. As already mentioned, human vision is wide-angle and low resolution for most of the visual field. Many visual systems in nature share these properties, as they favor navigation and self-defense. Foveal vision developed later in some vertebrates (including humans) for more specialized tasks that demanded high-resolution vision, but peripheral vision remains sufficient for many visually guided tasks (Navarro, 2009).

In the absence of foveated sensors, some architectures realize the distinction between peripheral and central vision using separate cameras. For example, Cog, a humanoid robot with movable head and arms, has two cameras per eye. One is for peripheral vision, providing a 88.6 × 115.8° FOV. Another type of camera is a high-resolution simulated fovea with narrow 18.4 × 24.4° FOV (Scassellati, 2003). Kismet, a baby-like successor of Cog, also has foveal narrow-angle cameras in each eye for face recognition and visually guided tasks. Two wide-angle peripheral cameras support search tasks, tracking and compensating for involuntary ego-motion (Breazeal et al., 2000). A non-humanoid robotic vehicle controlled by RCS uses a camera with 40° FOV and 242×256 px for central vision for object recognition and a peripheral camera with 115° FOV and same resolution to compute optical flow (Coombs et al., 1998). The iRobot B21R platform used for ACT-R/E (E stands for embodied) is equipped with omnidirectional camera and a high-resolution forward-facing camera (Kennedy et al., 2009).

Eye and head movements. Head and eye movements perform many important functions that make vision more versatile (e.g. by expanding the visual field and focusing on particular elements of the scene) and serve as non-verbal communication signals (e.g. gaze following, nodding). Many robotic platforms are equipped with pan/tilt heads that allow camera rotation. For example, 3T uses a pan/tilt unit to keep the person within the camera's FOV (Wong et al., 1995), an RCS-controlled robot performs saccades to realign the camera with its heading (Coombs et al., 1998). Kismet, mentioned earlier, can move its foveal cameras, but the peripheral camera remains stationary (Breazeal et al., 2000).

Range sensors

Although cameras are a rich source of information, existing algorithms for processing visual data are computationally expensive and unreliable. Various non-visual sensors can supplement or even replace cameras for some visual tasks, particularly those relying on accurate depth estimation. The three most common sensor types for this purpose are *infrared*, *sonar*, and *laser*. These remote sensors provide accurate distance measurements to objects using different mediums (infrared light, sound waves, and laser beams, respectively).

Strictly speaking, range sensors do not have the properties of human vision but rather provide better depth estimates than are possible from stereo or motion, and improve the overall robustness of the system. Depth cameras were introduced relatively recently and are used only by a handful of architectures. Such cameras cast modulated illumination and measure its distortion or travel time to compute depth of each pixel (Xiong et al., 2017).

Virtual sensors

Virtual sensors are typically part of the simulated environment and provide visual information in various forms, from unprocessed images or video streams to fully parsed scene representations, as will be discussed in the next section.

4.3.2 Stages of visual processing

The human ability to comprehend complex visual information appears effortless and instantaneous, but is a very complex biological process with multiple steps, many of which are still not well understood. Once light passes though the eye it falls on a sheet of photo receptors (retina in the eye or digital sensor in the camera) forming a representation of the scene. In the retina, however, this representation is analog and time-varying, unlike the static 2D snapshot of the digital camera. The raw visual information is of little use for most tasks and has to be processed further.

According to one of the early computational theories of visual processing by Barrow et al. (1978) and Marr (1982), three distinct stages follow: early, intermediate, and late processing. Early vision is data-driven and involves parallel processing of the visual scene and extracts simple elements, such as color, luminance, shape, motion, etc. Intermediate vision groups elements into regions, which are then further processed during the late stage, where individual objects, their locations, and spatial relationships between them are identified.

In the last forty years, our understanding of visual processing has been greatly improved, but the general division of visual processes into early, mid-, and late stages remains commonplace. We base our analysis on the taxonomy of image understanding stages introduced by Tsotsos (1992) as it maps well onto most of the cognitive architectures in our selection:

1. Detection and grouping of intensity-location-time values into edges, regions, and optical flow vectors;
2. Further grouping of edges, regions, etc., to infer surfaces, volumes, boundaries, and depth information;
3. Identification of the objects and their motion;
4. Building object-centered representations for entities;

5. Assigning labels to the objects based on the task;
6. Inference of spatio-temporal relationships among entities.[1]

All processing stages benefit from or even require additional task or world knowledge. Already at stage 2, grouping of features depends on the camera viewpoint and the kinds of objects present. Later stages require spatial reasoning and operate on representations abstracted from the results of earlier processing stages. Although not mentioned by Marr, visual attention mechanisms, emotion, and reward systems also influence all stages of visual processing (Tsotsos, 2011). Thus, perception and cognition are tightly intertwined throughout the entire visual pipeline.

In cognitive architectures, cognitive and biological fidelity of visual processing correlates with *theory conformity* (as defined in Section 3.2). Since none of the architectures have general purpose vision, most are designed for specific tasks. As a result, the type of *sensor input* and *task* affect what stages of visual processing they have and how they are implemented. For example, simple visually guided navigation can be done with very little processing since one needs to know the locations of obstacles, not their identities or properties. However, more challenging tasks, such as object manipulation and interaction with people, typically require most of the processing stages outlined above.

Cognitive inspiration and plausibility

Recalling the previous section, cognitive conformity constrains the kinds of representations, data structures, and algorithms that can be used in implementation. Typically, architectures that are merely informed by cognitive and biological theories are the least constrained and approach vision as an engineering problem, while still following the processing steps outlined above.

Early vision (step 1) usually involves edge detection and disparity estimation from stereo. These features are then grouped (step 2) into blobs with similar features (color, depth, etc.), which are resolved into candidate objects with centroid coordinates (step 3). For example, in the robotic systems it is common to use stereo images together with range sensors to detect obstacles without identifying them (e.g. ATLANTIS, Miller and Slack, 1991; CARACaS, Huntsberger et al., 2011).

To identify objects (step 4) and categorize them (step 5) it is common to use machine learning techniques. In this case, object models are learned off-line from a set of examples. For example, RCS (Albus and Barbera, 2006), DIARC (Scheutz et al., 2005), Kismet (Breazeal and Scassellati, 2002) use pretrained neural networks for object detection, DIARC (Schermerhorn et al., 2006) relies on SIFT features for object recognition, and CORTEX (Romero-Garcés et al., 2015b) trains SVM classifiers on stereo depth and LBP to find people and determine their age and gender. Hand-coded representations for object models are also common and require no learning.

Spatial reasoning (step 6) is often implemented as an ego-centric map that is populated with surrounding objects and regularly updated with new sensor information. Some robotic architectures feature an ego-sphere for integrating sensory information with action. This structure mimics the functions of the

[1]The last stage from Tsotsos (1992)—forming consistent internal descriptions—is omitted. In cognitive architectures, such representations are not confined to perception but are usually distributed across various reasoning and memory modules.

hippocampus, although not in a biologically plausible way (Peters et al., 2001b). Essentially, an ego-sphere is akin to a virtual dome surrounding the agent, onto which salient objects and events are mapped. RCS (Albus, 1994) and IMA (Kawamura et al., 2008) both implement this concept.

Adding more psychologically and biologically accurate processing is often challenging, because there are no universally accepted theories of biological vision nor off-the-shelf software that the less constrained architectures can use. As a result, those architectures that adhere more closely to biological realism are limited in terms of visual tasks they can perform.

More biologically plausible design of perception is based on the anatomy of the ventral pathway and models areas of the cortex that roughly correspond to early and intermediate processing steps described earlier. One prominent example is Leabra (O'Reilly et al., 2013), which organizes neurons in a hierarchy, models reciprocal connections between the layers, and recurrent inhibitory dynamics that limit the activity levels across the layers (Wyatte et al., 2012). The visual systems of Darwin VIII (BBD; (Seth et al., 2004b)), Spaun (SPA) (Eliasmith et al., 2012), and many ART models (Grossberg, 2007b) are also modeled on the primate ventral visual pathway. Neuron-based but not necessarily biologically realistic approaches to vision are implemented in SASE (Zhang et al., 2002), MDB (Duro et al., 2010), BECCA (Rohrer et al., 2009), and DAC (Mathews et al., 2009).

Although many of these systems do not explicitly perform tasks typical for late vision, such as assigning labels to objects, they are evidently capable of distinguishing between different object categories and their spatial relations for use in the visually guided tasks like navigation (BBD, Fleischer and Edelman, 2009; BECCA, Rohrer et al., 2009; DAC, Mathews et al., 2009; MDB, Duro et al., 2010; SASE, Weng, 2007).

Processing of real and simulated input

The source of visual input determines many of its characteristics and how much processing is required. Physical sensors (cameras, laser arrays, etc.) typically provide an *unprocessed* stream of sensor data (e.g. pixels, point clouds). While some advanced simulations can emulate physical sensors fairly accurately, more often they provide data that is *partially* or *fully* processed, i.e. includes additional information supplied by the simulation engine, such as object labels, locations, and other properties. Below, we discuss these different types of data with examples.

Unprocessed data. Some simulations render a stream of 2D arrays of pixels that represent images of the scene from the viewpoint of the agent. This is the closest one can get to a physical camera in simulation. However, only a few architectures use simulations that provide complete 3D scenes and cameras that move around to capture images from different viewpoints, similar to how a physical camera or the eyes of living organisms senses the surrounding environment. Even when such cameras are available, their parameters are rarely reported and presumably fixed.

There are several advanced simulations used for developing cognitive architectures. For example, MECA architecture controls a digital replica of a robot with stereo camera moving around a factory floor (Gudwin et al., 2020b). The robot controlled by CARACaS is situated in a game-like simulation of a Moon

base with multiple astronauts and objects (Huntsberger, 2011a). Similarly, the authors of CERA-CRANIUM use an avatar for the popular robotic platform Pioneer 3 with a single forward-facing camera, for which only the resolution (320×200 px) is specified. The environment is an indoor maze with textured floor and walls (Arrabales et al., 2009d). Another example, BECCA, uses a top-down view of the table with the robotic arm and the object it is learning to grasp. However, besides several screenshots of the simulated scenes, there is little information about the visual properties of the simulations.

Partially processed data. Many simulations instead of raw pixels provide partially processed data, e.g. objects with labels and other attributes that could be derived from the image, thus omitting explicit early and middle processing stages. Visual realism of such simulations is mostly for aesthetic and debugging purposes, as the information is available directly (sometimes with simulated sensor noise). For example, MIDAS implements an anthropomorphic model of the agent, cockpit, and external environment used mainly for visualization purposes (Hart et al., 2001). 3T operates within the kinematic simulation of the robotic arm for the tasks around a space station (Bonasso et al., 1997). MicroPsi receives visual shapes as shock graphs (skeleton and edges) from which it can classify objects. The ACT-R vision module retrieves information regarding color, location, and size of objects, from which spatial relations between them can be inferred (Byrne, 2001). EPIC follows a similar approach (Kieras and Hornof, 2014).

The grid-like 2D simulation environments, mentioned earlier in Section 4.2, are also commonly used by the symbolic cognitive architectures. Blocks World and similar simulations have been used by PRODIGY, Meta-AQUA, Soar, ICARUS, GLAIR, and Disciple.

Some simulations provide other types of sensor readings. For example, a simulated underwater vessel controlled by Clarion receives information from sonar gauge, range gauge, fuel gauge, and bearing gauge to detect mines (Sun and Peterson, 1998a) and a mobile robot used with the FORR architecture relies on the simulated sonars to detect obstacles during navigation.

Fully processed data. In this case, all necessary information is provided in a form easily digestible by the architecture, for example, as a problem statement in Lisp or any other internal representation. It is often difficult to determine what single sensory modality or the combination of several modalities the data could have come from, therefore we categorize such cognitive architectures as not implementing sensation and perception (Section 4.7).

4.3.3 Simplifying visual processing

A visual pipeline is often difficult to implement even for the simplest tasks due to sensor quality issues, inherent noise in the data, unpredictability of the environment, and limited processing resources. To counteract these problems, it is not uncommon to simplify visual processing with the following assumptions:

Uniform backgrounds and contrasting objects. Figure-ground segmentation and object identification become significantly easier if objects and areas lack textures or reflections, are uniformly colored, and are distinct. For example, the visual module built for the ISAC robot assumes that the objects of interest are light and placed against a dark surface. The face tracking system

also expects the hair of the users to be darker than the background (Kawamura et al., 1995). As the authors themselves admit, such conditions are restrictive in practice, especially when the users are not aware of them. Even some recent cognitive architectures, such as CERA-CRANIUM, expect objects to appear against a dark background (Arrabales et al., 2011).

Uniformly colored objects. Color-coding objects is another common practice that greatly simplifies object segmentation, identification, and tracking. In the ADAPT architecture, this approach is used to detect and track a red ball rolling on a featureless white table (Benjamin et al., 2013). A BBD-controlled robot likewise implements a very simple neural architecture for detecting and following a bright yellow object moving in the scene (Chen et al., 2013). In a more complex scenario, the robot controlled by DAC traverses a specially built maze by identifying color blobs corresponding to different targets: a yellow home location, blue and red blocks representing resources, and green landmarks (Maffei et al., 2015). Since human skin tone has a well-defined range and is unique among many other objects, tracking skin-colored blobs is a common solution for detecting people or their faces (e.g. 3T, Wang, 1995b; Kismet, Aryananda, 2001). However, skin detection techniques (and other color-based methods) may still be susceptible to illumination, clutter, and individual factors (Kakumanu et al., 2007).

Fiducial markers. Tagging objects with fiducial markers is a common way of making visual processing more robust, even in cluttered environments. Fiducial markers for computer vision are quick response (QR) codes represented as 2D binary patterns on a white background. These markers are designed for fast identification and can carry information about the location, orientation, and properties of the objects associated with them. Fiducials are easy to use as they can be generated using one of many off-the-shelf libraries, printed on a piece of paper, and attached to objects in the scene or rendered in simulation. As such, they are quite common. For example, MECA relies on fiducials to identify objects on the factory floor (Gudwin et al., 2020a), CORTEX uses them to localize itself in space (Romero-Garcés et al., 2015b), and Soar extracts object predicates from them (Mininger and Laird, 2016). Markers can also be used to identify people: an ACT-R model of infant gaze following relies on tracking the head of the user with a QR code (Trafton et al., 2009) and CARACaS uses markers to identify the astronauts and their locations in a simulated space mission scenario (Huntsberger, 2011a).

Top-down view. An overhead view, in addition to egocentric perspective, helps simplify problems related to 3D vision and navigation. This approach has been used in many architectures for a variety of tasks: TCA for finding objects in the room (Simmons, 1989), BECCA—to aid in grasping (Rohrer, 2007b), DUAL—for localizing the robot in the environment (Kokinov et al., 2008), FORR—to find other agents (Epstein et al., 2012), and Soar—to manipulate objects on the table (Laird and Mohan, 2014).

Additional sensors. Additional non-vision sensors can make visual processing more robust or even replace it altogether. For example, to more reliably detect static and moving objects around the robot, a camera can be augmented with infrared sensors, as is done in IMA (Rogers and Wilkes, 2000) and 3T (Kortenkamp et al., 1998). Similarly, range sensors provide an attractive alternative for tasks like autonomous navigation, where speed and quality of

the output are key. Thus, in the RCS-based autonomous vehicle, slow and costly stereo vision was replaced by LiDAR to perform early detection of drivable surfaces (Coombs et al., 2000).

Rigid geometry and distinctive features. This assumption is particularly important for object detectors, which often rely on keypoints (e.g. SIFT). Curved, non-rigid, and featureless objects/areas are much harder to identify, as was demonstrated in experiments with DIARC (Scheutz et al., 2007).

Good lighting conditions. The majority of the studies and demos found in the literature are conducted in good lighting conditions. Due to sensor noise and lack of training data, low light conditions often lead to poor results and require additional heuristics to make the output of the visual processing module usable for downstream modules. For example, face detection in low light conditions implemented for SS-RICS suffered from a high number of false positives. Thus, an additional template matching step was added to filter them out (Kelley et al., 2011).

4.4 Somatosensation, touch, and vestibular sense

There are several sensory modalities responsible for the sense of one's body; proprioception and balance keep track of the body parts, while touch and pain provide information about physical contact with external objects.

Most cognitive architectures, especially those embodied in hardware, implement proprioception, making it the second most common sensory modality after vision. Proprioception, or sensing the state of the body and its motion, has many uses, primarily in moving the actuators, navigation, and self-localization in the environment, while touch is necessary for grasping and manipulating objects. These senses play a role in social interactions as well, e.g. by helping position oneself appropriately in relation to others, for directing attention of others, and affective communication. In living organisms, information on skin deformation and the movements of extremities these senses are transmitted to the brain by a network of mechanoreceptors within skin, muscles, and joints. In artificial systems, a number of physical and virtual sensors are used to provide similar functionality: Global Positioning System (GPS) and compass for location and direction of motion, force sensors, and joint feedback for fine-grained information on the position and movement of joints, and whiskers or touch-sensitive bumpers for detecting contact with objects or surfaces.

Sensing pain, or nociception, is also part of somatosensation. This vital ability that nearly all biological organisms possess helps respond to physical damage and avoid further injury. While there is no organ responsible for pain in humans, the sensation of pain can result from specialized transducers, processing in the areas of the brain responsible for affective activities, or both (Loeser and Melzack, 1999).

Despite the importance of physical pain or discomfort for self-preservation, only a few architectures explicitly include a corresponding sense. Dedicated sensors for pain are particularly rare, with HCA as the only example, where specialized sensors detect hard contact or mechanical damage and emulate experiencing pain by disrupting attention to the current task (Haikonen, 2007).

Alternatively, other sensory modalities can be used to mimic pain. In robotic architectures, somatosensation plays a large role in protecting the

system from damage: joint feedback is important for checking whether the planned motor action is within a safe range of motion and touch-sensitive bumpers automatically stop the robot after hitting an obstacle. Perceptual signals can also be interpreted as nociception within the motivational system. In Kismet, high-intensity perceptual signals cause discomfort and elicit protective responses; the robot closes its eyes and rotates ears in response to bright light and loud sounds, respectively (Breazeal, 1998b). The MDB architecture interprets specific auditory signals given by the human teacher as pain or pleasure to be used during learning as reinforcement (Bellas et al., 2006). In RCS, pain is part of the value judgment module that evaluates and attaches a corresponding label to the perceived objects and events (Albus, 1996). Similarly, in the LIDA architecture, pain is managed by the motivational system which assigns the corresponding somatic markers to percepts in the early stages of sensory processing (Franklin and Ramamurthy, 2006). In both cases, pain can bias perception and subsequent actions. The integration of pain in the emotional system has also been proposed for Soar (Henninger et al., 2002) and ACT-R (Cochran et al., 2006).

Unlike touch, proprioception, and pain, vestibular sense is hard to describe or localize, but it provides essential information, such as direction of gravity, tilt, and acceleration of the body/head, that are needed for spatial cognition, perception of self-motion, and maintaining balance (Day and Fitzpatrick, 2005). These functions are not investigated in cognitive architectures we reviewed because nearly all of them are interfaced with non-humanoid bodies or do not possess a body at all. The only two instances where vestibular sense was considered are Cog (Brooks et al., 1999) and its successor Kismet (Breazeal, 2003b), both implementing vestibular-ocular reflex (VOR) and opto-kinetic response (OKR). The main purpose of VOR is to stabilize eyes during head motion. In Cog and Kismet, feedback from the 3 degree-of-freedom inertial sensor representing head movement is used to move the eyes in the opposite direction to maintain steady gaze. OKR supports VOR during slow head motions by generating eye movement in the direction of visual motion. In Cog and Kismet, OKR is simulated by measuring the optical flow of the background (visual slip) to calculate visual displacement and compensate for it.

4.5 Audition

Audition is another common modality, whose purpose is primarily to let human users interact with or guide the intelligent system using sounds or spoken commands. As such, the auditory modality is not deeply integrated with other elements of the perceptual system and is often implemented using off-the-shelf software for speech-to-text translation. Some examples are: greeting human users showcased with ISAC robot (Peters et al., 2001b), receiving directions to aid in object identification demonstrated using Polyscheme (Trafton et al., 2005), scripted interactions between a salesman robot controlled by CORTEX and human customers (Romero-Garcés et al., 2015b), and ordering books from the public library by phone with FORR (Epstein et al., 2011).

Most cognitive architectures focus on the linguistic and semantic information carried by the speech and only a few consider its other properties, such as loudness, speech rate, and intonation. These aspects of speech are

particularly important for social interactions and have been effectively used in several projects. For instance, Kismet robot categorises utterances as approving, prohibiting, or soothing based on the prosodic contours of the speech without any natural language processing (Breazeal and Aryananda, 2002). The Ymir architecture combines a prosody analyzer with a grammar-based speech recognizer with a limited vocabulary of a hundred words, for more natural turn-taking during conversation (Thórisson, 1999). The BBD robots can orient themselves toward the source of a specific auditory cue (Seth et al., 2004b) and MBD learns to associate the meaning of musical notes given as commands by the instructor (Bellas et al., 2006).

Among the few architectures that model auditory perception are ART, ACT-R, ARCADIA, SPA, and EPIC. For example, two instances of ART, ARTWORD and ARTSTREAM, were used for studying phonemic integration (Grossberg and Myers, 2000) and source segregation, also known as the cocktail party problem (Grossberg et al., 2004), respectively. A model of music interpretation was developed with ACT-R (Chikhaoui et al., 2009).

4.6 Olfaction and gustation

Both olfaction and gustation are important for ingestive behaviors, for example, perception of flavors, identifying whether food is suitable for consumption, and modulation of appetite (Stevenson, 2010). Smells have associated social aspects as well: the presence or lack of odor correlate with cleanliness, and fragrances signal social status or induce an emotional response. However, because foraging and social behaviors are rarely modeled in cognitive architectures, only a handful of them possess the ability to smell and taste.

Two architectures, GLAIR and MBCA, simulated the sense of smell. In the case of GLAIR, smell is one of the numeric parameters provided to the agent by the simulation to signal the proximity of danger (Shapiro and Kandefer, 2005). MBCA utilizes smell in a search-and-rescue scenario, where the scent of deodorant helps detect a hiker lost in the woods (Schneider, 2019).

Several architectures make use of physical sensors. DAC architecture supports chemosensors in a synthetic ant model that uses the sense of smell to locate the feeder and navigate toward it (Mathews et al., 2009). In BBD, the sense of taste is implemented via conductivity of the material that the robot touches when exploring the environment (high and low conductivity are assumed to be good and bad taste, respectively). Through operant conditioning, the robot gradually learns to associate certain visual features with taste, much like living organisms (Edelman, 2007). The water recovery system operated by 3T presents yet another version of gustation. Here, the sensors help identify the quality of the purified water for the crew(Bonasso, 2001).

4.7 Other input types

There is a large group of architectures that do not implement sensation or perception as such. These architectures instead receive input that represents the result of both processing stages and may represent more than one sensory

modality. The input may be in the form of text commands, a binary stream of data, or via graphical user interfaces (GUI).

Text input is typical for the architectures performing planning and logical inference tasks (e.g. NARS, Wang, 2013; OSCAR, Pollock, 1993; Disciple). Text commands are often written in terms of primitive predicates used in the architecture, therefore no natural language understanding is required.

Data input in the form of binary or floating point arrays is primarily used for the categorization and classification applications and may represent raw sensor data or features extracted from it (e.g. ART; Carpenter et al., 1991).

GUI are mainly used in human performance research to simplify the input of the expert knowledge and to allow multiple runs of the software with different parameters (e.g. IMPRINT, Mitchell, 2009; MAMID, Hudlicka et al., 2000; OMAR, Deutsch and Cramer, 1998).

4.8 Multimodal perception

So far, we have considered each sensory modality in isolation, but, in reality, the human brain receives a constant stream of information from different senses and integrates it into a coherent world representation. Multiple sensory modalities in biological and artificial systems improve the robustness of perception through complementarity and redundancy. However, using many sensors also introduces a number of challenges, such as incomplete, spurious, or conflicting data coming from different sensors, combining data with different properties (e.g. dimensionality or value ranges), the need for data alignment and association, etc.

Nearly half of all cognitive architectures can receive and combine input from more than two different types of sensors. This process is termed feature integration in cognitive science (Zmigrod and Hommel, 2013) and sensor data fusion in robotics (Khaleghi et al., 2013). Whether the sensors are part of the same modality or two or more different modalities, there are two main approaches to combining them:

Early integration. Data from various sensors is processed independently and combined as soon as possible into a single coherent picture. A common example we described earlier is an ego-sphere—a 3D map centered on the agent that assigns egocentric coordinates to various objects and properties of the environment based on the perceptual information from multiple sensors. This information can be used directly for tasks, such as navigation (e.g. CARACaS, Elkins et al., 2010; ATLANTIS, Gat, 1991a; RCS, Schlenoff et al., 2005). Additional modalities can be mapped onto this representation. For example, for the social robotics applications, audio information can be associated with the objects identified in the ego-sphere. This can be helpful for orienting the robot with respect to those interacting with it (IMA; Peters et al., 2001a).

One can also learn the association between the readings of different sensors without an explicit egocentric representation. For example, DAC uses Hebbian learning to establish data alignment for mapping neural representations of different sensory modalities to a common reference frame, mimicking the function of the superior colliculus of the brain (Mathews et al., 2012). The ARTMAP network integrates visual and ultrasonic sensory information via neural fusion for mobile robot navigation (Martens et al., 1998a).

Late integration. Instead of performing explicit data association and alignment, sensor data and feature extraction can be done independently and concurrently. Then, extracted information is directly added to a temporary memory storage (see Section 5.1.2). Any ambiguities and inconsistencies are resolved later during decision-making and reasoning. This is a common approach in distributed architectures, where independent modules concurrently work toward achieving a common goal (e.g. CERA-CRANIUM, Arrabales et al., 2009d; Polyscheme, Cassimatis et al., 2004; RoboCog, Bustos et al., 2013; Ymir, Thórisson, 1997; LIDA, Madl and Franklin, 2015).

All approaches mentioned so far rely on spatial and temporal proximity and/or learning to combine and disambiguate multimodal data, largely ignoring cross-modal interaction. At the same time, many psychological and neuroimaging experiments conducted in the past decades suggest that sensory modalities mutually affect one another. A well-known McGurk effect discovered by McGurk and MacDonald (1976) was one of the first demonstrations of how vision alters auditory processing. Consequent studies showed that interactions between sensory modalities are common and highly context dependent (Shimojo and Shams, 2001).

The only elaborate model of biologically plausible multisensory integration has been implemented in the BBD architecture. One of its instances, Darwin XI, was constructed to investigate the integration of multisensory information (from touch sensors, laser, camera, and magnetic compass) and the formation of place activity in the hippocampus during maze navigation (Fleischer et al., 2007). The neural network of Darwin XI consisted of approximately 80,000 neurons with 1.2 million synapses and simulates fifty neural areas. In a lesion study, the robustness of the system was demonstrated by removing one or several sensory inputs and remapping of the sensory neuronal units, which is consistent with the numerous studies on cross-modal plasticity in humans and other organisms (Shimojo and Shams, 2001).

4.9 Perceptual attention

Attention is one of the most studied aspects of human cognition, with thousands of papers on this topic published to date. Attention permeates cognition from perception to decision-making. However, it is not a monolithic structure, but a large and diverse set of mechanisms distributed across perceptual and cognitive processing pipeline, as ample evidence from psychology and neuroscience suggests (Tsotsos, 2011).

4.9.1 Visual attention

Just as vision is the most studied sensory modality, visual attention is the most researched and implemented form of attention. According to the comprehensive taxonomy proposed by Tsotsos (2011), attentional mechanisms can be grouped into three types of information reduction: selection, restriction, and suppression. In short, selection and restriction mechanisms reduce the search space by choosing relevant elements from many present. Suppression mechanisms work in the other direction and inhibit irrelevant elements.

Selection

Selective attention mechanisms help choose what to focus on. It is the most common type of visual attention and one with many existing implementations. The following mechanisms are typically considered selective:

World model. A world model defines the objects and events of which the agent is aware. Some sort of world model is included by default in virtually any cognitive architecture, even those with minimalist perceptual systems.

Feature/region/object/event of interest. Human attention is a serial process that moves from one object or area in the scene to another, depending on the task and surroundings. The selection of visual areas to attend to can be data-driven (also known as bottom-up) or task-driven (top-down).

The bottom-up attentional mechanisms identify salient regions whose visual features are distinct from the surrounding image features, usually along a combination of dimensions, such as color channels, edges, motion, rarity, etc. There are a number of bottom-up visual saliency algorithms that define which areas in the visual scene are more likely to attract attention. Implementations of these algorithms can be found in several cognitive architectures: e.g. Guided Search (Wolfe, 1994) is part of vision modules of ACT-R (Nyamsuren and Taatgen, 2013b) and Kismet (Breazeal and Scassellati, 1999), the Itti-Koch-Niebur saliency model (Itti et al., 1998) is used in ARCADIA (Bridewell and Bello, 2015) and DAC (Mathews et al., 2012), and model of attention based on information maximization (AIM; Bruce and Tsotsos, 2005) is part of the early implementation of STAR (Kotseruba, 2016).

Top-down selection can further limit sensory data provided by the bottom-up processing. For example, in visual search, knowing desired features of the object, such as its color, can help reduce the number of candidates provided by the data-driven figure-ground segmentation. Many architectures resort to this mechanism to improve search efficiency (ACT-R, Salvucci, 2000; ARCADIA, Bello et al., 2016; CERA-CRANIUM, Arrabales et al., 2009d; DAC, Mathews et al., 2012). Another option is to use a hard-coded or learned heuristics. For example, CHREST looks at typical positions on a chess board (Lane et al., 2008) and MIDAS replicates common eye scan patterns of pilots (Gore et al., 2009). The limitation of the current top-down approaches is that they can direct vision for only a limited set of predefined visual tasks, however, ongoing research in STAR aims to address this problem (Kotseruba, 2016; Rosenfeld et al., 2018; Tsotsos et al., 2018).

Viewpoint/gaze selection. These two mechanisms perform complementary functions: gaze control brings a portion of the environment into the central view of the camera, and viewpoint selection helps get more information about the region/object by moving closer to it or by viewing it from a different angle. Both are needed for visual exploration, but only viewpoint selection is common. Since the majority of cognitive architectures use visual sensors with limited FOV, they automatically support viewpoint selection because the sensor has to be moved to capture the relevant parts of the environment.

Eye movements that occur within the viewpoint and are independent of the body movement are more difficult to achieve. One needs to implement foveal and peripheral vision, either programmatically or physically. Programmatic methods include the application of anisotropic blur (as in STAR; Kotseruba,

2016,Wloka et al., 2018) or selective processing of the image (CERA-CRANIUM; Arrabales et al., 2009d). In simulated environments where no image data is available, one can artificially limit the amount of information to simulate differences in foveal and peripheral perception. This is done in EPIC (Kieras and Meyer, 1996), ACT-R (Nyamsuren and Taatgen, 2013a), and MIDAS (Tyler et al., 1998). The physically foveated systems use pairs of cameras with different fields of view and resolution, as discussed earlier in Section 4.3.1.

Restriction

Restriction mechanisms reduce the visual processing complexity by limiting the search space to certain features (priming), events (exogenous cues), areas (visual field), objects (exogenous task), and knowledge (endogenous motivations). The exogenous task and endogenous motivations are implemented in virtually all architectures by default as task instructions and domain knowledge, respectively. Attention can also be modulated by internal signals, such as motivation or emotions (discussed in Chapter 7).

A limited visual field is a feature of any physically embodied system, since most cameras do not have a 360° FOV. This feature is also present in some simulated vision systems. The ability to react to sudden events (exogenous cues), such as fast movement or bright color, is common in the social robotics applications (e.g. ISAC, Kawamura et al., 2004; Kismet, Breazeal et al., 2000).

Priming is another mechanism that allows biasing the visual system toward particular types of stimuli via specific task instruction. For example, human detection can be improved by assigning more weight to the skin-colored features during the saliency computation (Kismet; Breazeal et al., 2001). One can also use the knowledge of the probable location of some objects to spatially bias their detection (STAR; Kotseruba, 2016, Rosenfeld et al., 2018). Numerous ART models implement top-down priming and demonstrate its advantages in visual tasks and learning (Carpenter and Grossberg, 1987b).

Suppression

Suppression is complementary to selection and restriction mechanisms, as it acts to improve signal-to-noise within a receptive field by inhibiting elements that are not relevant to the task. Its impact is multiplied when implemented hierarchically. The selected stimulus then becomes the focus of attention and the rest is treated as noise.

Some of the most common such mechanisms are surround inhibition (temporary suppression of features around the attended object), inhibition of return (IOR)(temporarily prevents returning attention to previously focused locations or stimuli), and suppression of task-irrelevant stimuli.

Despite the large role that suppression plays in biological systems, only a handful of the architectures implement suppression mechanisms. For instance, the suppression of task-irrelevant visual information is done in only three architectures: in BECCA, irrelevant features are suppressed through the winner-take-all (WTA) mechanisms (Rohrer, 2011) and in ARCADIA non-central regions of the visual field are inhibited at each cycle since the cue always appears in the center (Bridewell and Bello, 2015). IOR is another suppression mechanism, which prevents the visual system from repeatedly attending to the same salient stimuli. In ACT-R, this mechanism is tied to the activation values

and distance, which are used to ignore the objects for consecutive "where" requests (Nyamsuren and Taatgen, 2013b). In ART and STAR, an additional map is used to keep records of the attended locations. ARCADIA also has a covert inhibition mechanism, however, its implementation details are not provided (Bridewell and Bello, 2015). IOR is also part of saliency algorithms mentioned earlier, thus architectures using these solutions include it as well.

Overall, visual attention is largely overlooked in cognitive architectures research except for the architectures that study attention (e.g. ART, ARCADIA, STAR). This is surprising because strong theoretical arguments for the importance of attention in dealing with the computational complexity of visual processing have been known for decades (Tsotsos, 1990).

Most often, attention mechanisms found in cognitive architectures are side effects of other design decisions. For example, task constraints, world model, and domain knowledge are implemented by default because they are needed for functioning of other aspects of the architecture. Limited visual field and viewpoint changes likewise often result automatically from physical embodiment. Otherwise, the only common mechanisms intentionally included for optimizing visual processing are region of interest selection and visual reaction to exogenous cues.

4.9.2 Auditory and other types of attention

Only a few architectures include attentional modulation for audition, but not other sensory modalities. Rudimentary mechanisms of auditory attention are implemented in OMAR (Deutsch et al., 2014) and MIDAS (Gore et al., 2009), where attending to an auditory stream is required for its comprehension. More advanced models of selective auditory attention are implemented in EPIC (Kieras et al., 2016) and ARCADIA (Gever et al., 2020). Both were used to investigate the "cocktail party problem." The setup, first described in (Cherry, 1953), describes a problem of identifying speech from one person among other conversations happening simultaneously and in close proximity.

In the cognitive and psychological literature, attention is used as a broad term for the allocation of limited resources (Rosenbloom et al., 2015b). For instance, in the Global Workspace Theory (GWT; Baars, 1988) attentional mechanisms are central to perception, cognition, and action. According to GWT, the nervous system is organized as multiple specialized processes running in parallel. Coalitions of these processes compete for attention in the global workspace, and the contents of the winning coalition are broadcast to all other processes. For instance, the LIDA architecture is an implementation of GWT[2] (Franklin et al., 2012). Other architectures influenced by the GWT include ARCADIA (Bridewell and Bello, 2015) and CERA-CRANIUM (Arrabales et al., 2009c).

Along the same lines, attention is often found in the architectures that simulate cognition as a set of autonomous processes (similar to Minsky's (1988) Society of Mind). Attention values for each process may be assigned directly or learned from experience, vary dynamically, and affect action selection and resource allocation. This idea is implemented in COGNET (Zachary et al.,

[2]In GWT, focus of attention is associated with consciousness, however, we do not discuss this topic in the book.

2000), DUAL (Kiryazov et al., 2006), and Copycat/Metacat (Mitchell and Hofstadter, 1990) in which multiple processes compete for attention. Other computational mechanisms are also possible: in Polyscheme a simple queue is used (Cassimatis et al., 2009), attractor networks in CogPrime (Ikle and Goertzel, 2011), and modulators in MicroPsi (Bach, 2015).

4.10 Summary

- Most cognitive architectures do not represent human sensory experience and perception on a deep level. None of the systems implement all basic senses available to humans, even in the simplest form.
- There is a large gap in how sensation and perception are approached in the architectures and modern views on this topic. Sensory processing in both older and more recent cognitive architectures remains vision-centric, despite accumulated evidence in favor of multisensory experience.
- Technical characteristics of the sensors used for individual modalities are often not available or do not represent what human senses can provide, even for the recent cognitive architectures. The use of sensors that either surpass or are inferior to human abilities is problematic since it compromises cognitive architectures with respect to their goals.
- Inclusion of a particular sense is dictated primarily by its function, i.e. a way of delivering the input to the system in a form sufficient for performing useful tasks and little else.
- Even though technically proprioception is the second-largest implemented modality, its role often is rather rudimentary and utilitarian, as a way to prevent damage to the equipment or aid in visual tasks.
- Multimodal perception is not very much studied. Although many robotic systems perform sensory fusion, its primary purpose is to increase robustness by pairing sensors with complementary properties.
- Attention permeates all aspects of human perception, especially vision, but is largely overlooked in cognitive architectures. In many cases, attentional mechanisms are not biologically motivated and are side effects of other design decisions necessary for functioning of other aspects of the intelligent system (e.g. world model, domain knowledge, task constraints).

5 Memory

Broadly speaking, memory is the ability to preserve information and retrieve it when needed. Anything can be remembered—facts, objects, events, skills, mental states, feelings, sensations, and more. Memory is integral to cognition because it plays a role in all cognitive abilities; object permanence during perception, keeping track of goals for decision-making, backtracking an argument when reasoning, and storing newly acquired knowledge are all enabled by memory.

Given its importance, memory is the only cognitive ability that all architectures possess. However, despite functional similarities, specific implementations of memory systems differ significantly depending on the research goals, theoretical underpinnings, and constraints—be they biological (e.g. neural plausibility) or engineering (e.g. limitations of programming language or software architecture). This chapter examines several key aspects of memory.

Section 5.1 is dedicated to memory types grouped by persistence, such as ultra-short-term (sensory), short-term (STM), and long-term memory (LTM). The central role of working memory (WM), its functions, properties, and dominant theories of how it operates are emphasized.

Section 5.2 discusses memory types categorized by contents. Here, the main distinction is made between declarative and non-declarative memory stores, which are widely accepted in the psychological literature and supported by the majority of cognitive architectures.

Section 5.3 argues for forgetting as a necessary function of memory rather than a failure to retain information.

5.1 Memory types by persistence

Hebb (1949) was the first to propose dividing memory into transient and long-term stores and describe the underlying neural mechanisms. However, the theory that gained widespread acceptance was Atkinson and Shiffrin's (1968) three-store model with the following components:

- *Ultra-short-term memory*—memory with the shortest span, also known as the *sensory register*, retains sensory information for <1 s;
- *Short-term memory (STM)* —temporary memory storage that retains information for up to 30 s and has limited capacity;
- *Long-term memory (LTM)*—the most persistent and the largest memory store, with potentially lifetime retention and unlimited capacity.

In this model, each of the memory systems plays a specific role in information processing. The sequence of processing starts from the sensory system, proceeds to the ultra-short-term sensory register,

into LTM. However, not all incoming sensory information arrives at LTM, as some may be lost at any stage.

The popularity of this model is partly due to its suitability for computational modeling since there is an apparent resemblance between three memory systems and parts of computer memory: sensory register, STM, and LTM map to computer registers, RAM, and hard drive, respectively (Mueller and Mueller, 1995). Many cognitive architectures also implement persistence-based separation of memory, although not always explicitly based on the Atkinson-Shiffrin model. In the remainder of this section, we will look into each of the three types of memory in more detail.

5.1.1 Sensory memory

Ultra-short-term or sensory memory makes sensory information available for a brief period after it no longer exists in the physical world (e.g. sound) or is no longer being sensed, primarily to enable sensory persistence. According to Cowan (2008), the first demonstration of this phenomenon and estimate of the duration of visual sensory memory (also known as iconic memory) dates back to 1740 experiments by the German physicist Johann Segner. He set up a dark room with a cartwheel to which a glowing coal chunk was attached. By rotating the cartwheel at increasing speeds, Segner determined that the observer could perceive a full circle when the wheel completed a rotation in 100 ms, suggesting this time as the limit of iconic memory. Sperling (1960) later used a tachistoscope (a device for timed exposure to stimuli) to more precisely measure the range of iconic memory at 250–1,000 ms. Similar buffers have been discovered for other sensory modalities as well; for example, echoic (auditory) memory, crucial for speech processing, persists from 2 to 5 s (Lu et al., 1992) and haptic memory which helps retain information about the recently picked up object for up to 2 s (Shih et al., 2009). Besides the ultra-short persistence, the following properties have been established for sensory memory (Werning and Cheng, 2017):

- Specialized memory stores are not shared across sensory modalities;
- The contents are composed of only external stimuli;
- There is little to no processing of the sensory data;
- Rates of memory decay vary depending on the sensory modality.

Several architectures implement sensory memory, but only a few elaborate on its structure and workings, and even fewer validate its properties and performance. The architectures, such as EPIC, LIDA, CogPrime, and RCS, support separate sensory memories. Of these, only the EPIC implementation closely followed psychological literature on defining the structure and functioning of its iconic and echoic sensory stores (Kieras et al., 1997). In EPIC, sensory information is retained for 200 ms after the stimulus disappears, however, within that time, different delays may be assigned to different properties of objects or sounds. There is no limit to the amount of information that can be held (Kieras and Meyer, 1996). EPIC's visual system has been validated against human data on a visual search task (Kieras and Hornof, 2014).

Several other architectures, such as Sigma, DIARC, ICARUS, ARCADIA, LIDA, and CogPrime, have perceptual buffers as well, but there is limited information on their contents, persistence, or capacity limits. LIDA, for example,

is described as having separate stores for sensory modalities (iconic, echoic, and haptic), and an additional store for integrating multimodal information (Franklin and Baars, 2010). ICARUS implements a perceptual buffer for holding descriptions of physical entities obtained from virtual sensors. This buffer updates on every cognitive cycle (Langley et al., 2005b). In ARCADIA, iconic memory overlaps in functionality with STM, as it not only holds consecutive frames from the camera but also performs segmentation to detect regions, their colors, and retinotopic locations (Bridewell and Bello, 2015).

Since sensory memory largely depends on external stimuli, it may be considered part of the perceptual system rather than memory. For example, some vision pipelines described in the previous chapter have dedicated buffers for storing sequences of images or other types of data from sensors. The main purpose of these buffers is motion detection through frame differencing or optical flow computation (e.g. in TCA, Kismet, and 3T). In most architectures, perceptual processing is also tightly linked with STM or working memory, which will be discussed next.

5.1.2 Working memory

The literature on WM is colossal. While there is an agreement that the purpose of WM is to temporarily hold and process information during task execution, the structure, contents, principles of operation, and interactions between WM and the rest of human memory system are still being debated (Chai et al., 2018). The same can be said about WM (or structures equivalent to it) in cognitive architectures. In this section, we will discuss the commonalities and differences in WM functions and implementations across many architectures, psychological motivation of computational models, and their validation against human data.

Working memory vs. short-term memory

Two common terms for temporary data store are STM and WM. Although two concepts are closely related and often used interchangeably, they are not the same; it is generally assumed that STM merely holds the information, while WM performs processing (Baddeley, 2012). However, according to the classic volume on memory by Priti and Miyake (1997) and later review by Aben et al. (2012), empirical evidence from psychology and neurobiology does not draw a strict boundary between the two, leaving room for multiple hypothetical models of their interaction. For instance, Aben et al. (2012) list seven plausible configurations of STM and WM, ranging from completely separate to fully overlapping, with several intermediary variants, where STM and WM interact or overlap only partially.

Given this ambiguity, it should not come as a surprise that there is no standard way of implementing STM and WM across cognitive architectures either. We will start our discussion by examining what types of temporary memory are available in different architectures and, if there is more than one, how they are related. Depending on how the terms "STM" and "WM" are used in the publications, we can identify five groups:

Neither term is used. A number of architectures are not described as having "short-term memory" or "working memory," but include structures that play similar roles. For example, blackboard-like architectures (e.g. PRS, FORR,

and Ymir) fall in this category because the blackboard maintains information relevant to the current task or long-term goals, effectively serving some function of both short-term and working memories. 3T uses the term "perceptual memory" that temporarily holds task-related objects located in the physical proximity of the agent (Wasson et al., 1999). Memory in NARS likewise stores knowledge temporarily and serves as a workspace for processing (Wang, 1995a), but is not referred to as STM or WM (Wang, 2012).

No "WM," only "STM." Only two architectures do not mention WM at all. In RCS, STM holds attended entities, pointers to related items, and recent events but does little processing. Functions of WM are performed by the world model (which includes STM) (Albus et al., 2002). In HCA, STM likewise stores percepts and the task-relevant information, but it is unclear what if any processing is performed (Haikonen, 2007).

No "STM," only "WM." Cognitive architectures, such as APEX, ADAPT, BB1, BECCA, CERA-CRANIUM, CogPrime, Companion, CORTEX, MA-MID, MBCA, MECA, SAL, SASE, Sigma, and SPA, implement only one memory component that maintains and processes task-relevant information.

"STM" ≠ "WM." Several architectures, including ART, MIDAS, BBD, IMA, DAC, SHRUTI, MDB, ARCADIA, and STAR, implement two temporary memories with different functionalities: one called STM is used to accumulate information, whereas processing occurs in WM.

"STM" == "WM". In the remaining architectures, "working memory" appears more frequently, but both terms are used interchangeably and considered the same.

In summary, some cognitive architectures do not use psychological terms for transient memory but have structures that temporarily store and manipulate task-relevant information. Those architectures that explicitly use the terms "working memory" or "short-term memory" generally consider both equivalent to one another and to WM as defined in the psychological literature. For consistency, we will adopt this convention and refer to temporary memory stores collectively as WM for the remainder of the section.

Overview of working memory

The purpose of WM is to maintain information that may no longer be accessible from the world, but is relevant to the goal(s) or task(s) being performed. As such, WM is central to human cognition, because it connects perception with the decision-making and reasoning abilities. In most cognitive architectures, WM (or components playing its role) is centrally positioned and well connected to other modules. Figure 5.1 schematically shows connections between WM, perception, LTM, decision-making, and attention. Below, we briefly describe the interactions between the components along the arrows:

Perception ↔ working memory: entities identified through perception are transferred to WM; in the opposite direction, contents of the WM can be used to influence perception.

Working memory ↔ long-term memory: information within WM serves as a cue for retrieving information from LTM that may be relevant for the current task or goal. Conversely, items from WM may move to LTM under certain conditions.

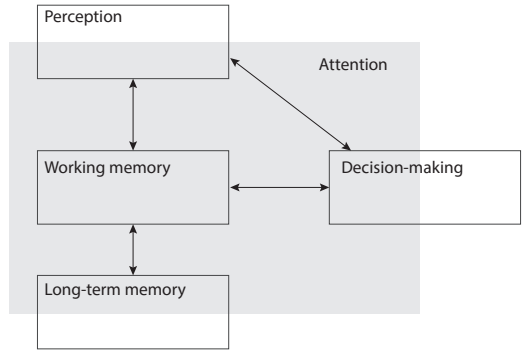

Fig. 5.1 A schematic diagram of the relationship among perception, working memory, long-term memory, and decision-making. The attention mechanisms (shown as a gray box) are involved along the arrows.

Decision-making ↔ perception: actions may involve changes to the sensors (e.g. turning the camera or moving in space), affecting what parts of the environment can be perceived. In reverse, certain stimuli can trigger immediate reflexive actions (e.g. avoiding imminent collision indicated by proximity sensors).
Working memory ↔ decision-making: the current goal/task determines what items will reside in WM; however, if the available information is not sufficient, additional goals may be created as a result.

Attention is integral to the interactions between the components of a memory system (Kiyonaga and Egner, 2013). In fact, attention and WM are so closely related that WM can be considered an attention system in its own right (Oberauer, 2019). Specifically, perceptual attention, discussed in Section 4.9, affects which percepts enter WM: bottom-up attention automatically selects percepts based on their saliency and top-down attention prioritizes certain items over others based on their relevance to the task. WM, in turn, affects attention and perception, regardless of relevance of memory contents to the task. A reciprocal relationship between WM and attention exists also for non-perceptual elements, such as concepts, goals, and actions (Oberauer, 2019). For example, attentional selection is involved in retrieval from LTM, choosing a response from available alternatives, and task-switching. By doing so, attention helps maintain focus on the current task and goal while avoiding irrelevant objectives. At the same time, attention can be subject to automatic capture by failures and distractors.

Models of working memory

Numerous theories of the WM mechanisms exist in the psychological literature, but only three models were particularly influential in the cognitive architecture domain: Baddeley's model (Baddeley and Hitch, 1974; Baddeley, 1986), the blackboard model (Hayes-Roth and Collinot, 1994), and long-term working memory (LT-WM; Ericsson and Kintsch, 1995). An illustration of which architectures implement which model is given in Figure 5.2.

Baddeley's model. The model of WM proposed by Baddeley and Hitch (1974) remains one of the most dominant, despite being nearly half a century

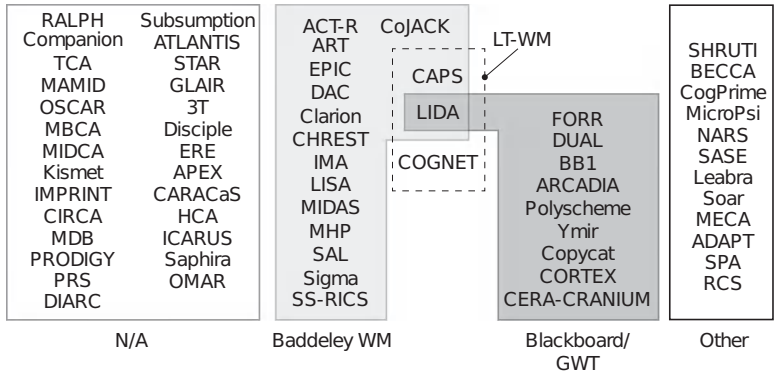

Fig. 5.2 Working memory implementations across architectures divided into the following categories: "N/A"—do not use the term or do not provide enough details regarding working memory implementation and theory, "Baddeley WM"—based on or compatible with the psychological model by Baddeley et al., "LT-WM"—implement or compatible with the long-term working memory model by Ericsson & Kintsch, "blackboard/GWT"—implement or compatible with the blackboard or Global Workspace Theory, "other"—implementations not directly related to the existing psychological models and to one another.

old. It extended an earlier Atkinson and Shiffrin (1968) memory model with more elaborate STM components to address experimental results related to verbal reasoning, comprehension, and free recall (repeating items presented earlier in any order). The following components were added: a phonological loop for maintaining auditory information through rehearsal and a visuospatial sketchpad for holding visual and spatial information. The third component, a central executive, was introduced to control what went into each independent memory store and to coordinate their processing. Due to the more active role of the central executive, the phonological loop and visuospatial sketchpad are sometimes referred to as "slave" subsystems. In the subsequent major revision, Baddeley (1986) reformulated the central executive based on Norman and Shallice's (1986) attentional framework, which differentiated automatic actions based on habits (schemata) and controlled deliberate actions that apply in situations where habits do not exist or are not helpful.

The most recent major change to this model was the introduction of the episodic buffer (Baddeley, 2000). It addressed the binding problem and chunking by combining the WM output with relevant content from LTM into episodes and mediating their transfer to and from LTM. A detailed discussion of model development, its explanatory power, and relationship to other theories can be found in Baddeley (2012).

Many architectures have been inspired by Baddeley's WM model or its individual components. For example, MIDAS, EPIC, and IMA incorporate the earliest version of the model with the phonological loop and the visuospatial sketchpad. Clarion and ART model their WM on Baddeley (1986), and Sigma has an episodic buffer inspired by Baddeley (2000). Although ACT-R does not implement Baddeley's model directly, it is generally compatible with it (Anderson et al., 1996), particularly in the implementation of the articulatory loop and emphasis on the time-based decay of WM elements (Anderson et al., 1998). CHREST has a visuospatial sketchpad as described in (Baddeley, 1986) implemented as a time-limited store for items that were perceived or

retrieved from LTM (Gobet, 2013). CHREST has also been augmented with the decay-based phonological loop described in Baddeley and Hitch, 1974 but without the rehearsal mechanism (Lloyd-Kelly et al., 2016). CAPS, on the other hand, implements only the part of the central executive responsible for language comprehension and does not include modality-specific buffers (Just and Carpenter, 1992).

Leabra is perhaps the only architecture whose authors are critical of Baddeley's theory. They model WM as an emergent property of neural networks operating on distributed representations, thus making a step toward eliminating central executive, which they see as problematic (O'Reilly and Munakata, 2000, p. 218).

Blackboard. A large group of architectures implements a blackboard-style WM. In the blackboard paradigm, multiple processes run in parallel and use the blackboard as a shared repository for goals, problems, and partial results (as described earlier in Section 2.2). An often invoked analogy for this style of processing is a group of people completing a jigsaw puzzle. Although the blackboard itself is not biologically plausible, it is similar in functionality to WM in production systems (Pfleger and Hayes-Roth, 1997) and Baddeley's model (Hayes-Roth and Collinot, 1994).

Examples of architectures that adopted this approach, besides the BB1 that introduced it, include Ymir, Polyscheme, and Copycat. In Ymir, three separate blackboards maintain communication between the processes for low-level perception, control, and motor actions (Thórisson, 1999). In Polyscheme, the focus of attention is modeled after the blackboard but with some differences. Focus of attention itself is accessible by multiple modules, but is very small and contains only the results of computations. Modules perform most of the processing locally, without sharing with one another, and can have arbitrarily large memories with arbitrary organization, unlike the original blackboard systems, where the blackboard was the main memory of the system (Cassimatis et al., 2009). In Copycat, the blackboard is called Workspace and performs functions of STM and WM. Workspace maintains perceptual items as well as relations between them and activations that accumulate as the solution to the problem is being developed (Hofstadter and Mitchell, 1994).

Robotic architectures, such as CORTEX and PRS, use blackboards primarily to enable real-time operation. CORTEX maintains all perceptual information and intermediate results in a Deep State Representation (DSR), which can be read and modified by the processes working on a problem. Specialized data types and asynchronous processing help maintain low latency and consistency of the memory as multiple agents read and write to it (Bachiller et al., 2021). PRS modifies the original blackboard control mechanism to ensure real-time execution by adding a meta-monitoring function to interrupt processes that take too long or rearrange the priority of tasks depending on the situation.

Baars's (1983) Global Workspace Theory (GWT) is similar to the blackboard concept in that it views brain activity as a collection of specialized processes interacting through transient storage called global workspace. While the blackboard theory did not operate with cognitive terms, GWT draws parallels with consciousness and neural processes.

An implementation of GWT in IDA and its successor LIDA realizes many of the functions of Baddeley's model framed in terms of the GWT theory

(Franklin et al., 2016). For instance, the slave systems are related to preconscious processing. Retrieval, repetition, and manipulation of items are a result of the competition between multiple independent processors that affect the activations of WM items. While the slave systems operate autonomously, they are guided by the executive functions (in this case, goal hierarchies). Only a single process can win at every cycle and broadcast its information to other processes, raising the activation of the corresponding items in memory.

Although not as elaborate, similar processes are employed in other architectures inspired by GWT, such as ARCADIA (Bridewell and Bello, 2015), MECA (Gudwin et al., 2018), and CERA-CRANIUM (Arrabales et al., 2009b).

Long-term working memory. LT-WM proposed by Ericsson and Kintsch (1995) gained influence by explaining how mnemonic techniques can effectively extend WM well beyond its typical capacity in common recall tasks. High recall accuracy was attributed to domain-specific stable structures in LTM and retrieval cues that kept these structures easily accessible in WM.

LT-WM is featured in CAPS, LIDA, COGNET, and LISA. In COGNET, it is the core model of WM, while in all others it is combined with the Baddeley-style WM. In LISA, LT-WM comprises the active subset of LTM, whereas WM contains the elements currently attended to (Kubose et al., 2002). LIDA likewise has a separate Baddeley-style memory (Franklin et al., 2013), which is different from an active LT-WM-like memory. CAPS implements an LT-WM model for the expert in digit span[1], which is used as an existence proof of the consistency of the capacity and the LT-WM theories (Just and Varma, 2002).

Although according to Baddeley's (2012) own admission, his theory and LT-WM are not contradictory, CHREST architecture presents a different view. Specifically, Gobet (2000) highlights a number of weaknesses of LT-WM as a verbal theory, including vague definitions for memory mechanisms, structures, and parameters, and weak generalizability to other domains. To prove the latter point, it is shown that LT-WM's predictions are not accurate in the case of recalling random meaningless chess positions, whereas the model based on CHREST fits data from chess experts better.

Working memory capacity

The limited capacity of WM is perhaps its most known and most studied property, following Miller's (1956) discovery of the "magical" number of 7 ± 2 chunks (or items of information) that an average person can hold in memory for several seconds. Later findings have reduced this limit to 3–5 items (Cowan, 2001), which is also the limit set in the Baddeley and Hitch's (1974) model.

Similar limits on WM exist in some architectures: CHREST and Clarion limit the default WM capacity at 4 chunks, IMA uses between 2 and 9 chunks depending on the task, and LISA sets memory capacity at 3 items. Others impose only temporal limits such that WM contents are accumulated for a set horizon from seconds to hours (PRS), tens of seconds (LISA), a few seconds (SPA), or less than 1 s (SASE). ACT-R, CAPS, MAMID, MIDAS, and DUAL limit capacity by the total allowed activation of all WM elements, which also effectively controls their number.

[1]Digit span is a common psychological task that measures how well people can memorize and repeat a sequence of numbers shown to them earlier. Variants of this task measure capacity and other aspects of WM (Conway et al., 2005).

In the remaining architectures, WM is effectively unbounded but for different reasons. For example, EPIC's authors motivate this decision by not willing to "hard-wire [. . .] collective ignorance," instead opting for experimentation to determine appropriate limits (Kieras, 2007). Others use implicit mechanisms that curb the infinite growth of WM contents by discarding or replacing the items based on use and recency. For example, 3T removes information not necessary for the current task, whereas in ARCADIA, BECCA, Copycat, and EPIC, WM elements gradually decay and eventually disappear if not used.

A more complex schema is implemented in Soar, where two types of permanence exist: elements created during operator application persist until they decay or are explicitly removed, while other elements are removed as soon as the production rule that instantiated them no longer matches WM (Nuxoll et al., 2004).

Psychological and biological realism

Although most WM implementations are based on psychological theories, only a small number of architectures can replicate psychological phenomena, and even fewer model the underlying neural mechanisms. Given WM's central role in many cognitive abilities and the vast range of specific effects it influences, there is minimal overlap across architectures in terms of tasks or functions of WM they model.

This can be seen in Table 5.1 which lists architectures that replicated human experimental data for various WM phenomena. ART and CHREST focus on the link between perception and WM, EPIC, Leabra, and SPA model the effects of WM capacity on classic serial memory tasks, and LISA investigates normal and impaired functioning of WM for analogical reasoning. Since the contents of WM are not observable, indirect measures, such as reaction times or error trends are used instead, although some architectures, such as ART, ACT-R, CAPS, and SPA, draw parallels between the activity in certain modules of the architecture and corresponding brain areas. As a result, it is difficult to assess relative advantages and disadvantages of different representations, a topic to which we will return in Chapter 10.

5.1.3 Long-term memory

LTM, as its name suggests, holds information for extended periods, potentially throughout an agent's lifetime. Historically, memory-related research in cognitive architectures concerned mainly organization and representation, which will be discussed next in Section 5.2. Relatively less attention has been paid to managing large-scale memories, partly because most cognitive architectures were only tested in simple domains and for limited amounts of time. However, understanding the challenges of maintaining large memory systems is crucial for matching human-like speed and accuracy of information retrieval in real-world tasks.

To set a reference point for memory systems in cognitive architectures, we will start with human memory. Most attempts to measure the size of human memory focus on the number of known words. While results vary depending on methodology, subjects, and units, recent studies suggest that adult English speakers typically know about 40,000 lemmas (uninflected words) on average (Brysbaert et al., 2016). In terms of storage, early estimates put memory

Table 5.1 Psychological and neural working memory phenomena modeled in cognitive architectures.

Architecture	Working memory phenomena
ACT-R	Individual differences in WM (Daily et al., 2001) Effect of task complexity and WM span (Anderson et al., 1996) WM and asymmetric costs of switching between tasks (Schneider and Anderson, 2010) Individual differences in WM and skill learning (Taatgen, 2020) Neural correlates of WM (Rosenberg-Lee et al., 2009; Anderson et al., 2016)
ARCADIA	Cognitive load and WM (O'Neill et al., 2018)
ART	WM for word perception (Grossberg and Myers, 2000; Grossberg, 2003) WM for speech perception (Cohen and Grossberg, 1986) Role of WM in sensory-motor planning (Grossberg and Merrill, 1992) WM for 3D visual object recognition (Bradski et al., 1992; Carpenter and Grossberg, 1993) WM for storing event sequences (Grossberg, 2007a)
CAPS	Individual differences in WM tasks (Newman et al., 2003) Model of LT-WM phenomena (Just and Varma, 2002) Aphasic sentence comprehension explained by reduction in WM activation (Haarmann et al., 1997)
CHREST	Effect of WM capacity on recall of chess positions (Smith et al., 2008) WM capacity and recall of random chess positions (Gobet, 1998)
DUAL	Impairment of WM by anxiety (Feldman and Kokinov, 2009) Visual WM for stimulus-driven attention (Nestor and Kokinov, 2004) WM for perceptual judgment (Kokinov et al., 2004)
EPIC	Verbal WM for serial memory-span tasks replicating multiple effects (sequence length, articulation time, phonological difficulty and articulatory suppression) (Kieras et al., 1998)
Leabra	Model of WM tested on a number of standard experiments (Hazy et al., 2006) Model of n-back task (Chatham et al., 2011)
LISA	Analogical reasoning involving WM in patients with dementia (Morrison et al., 2004) Analogical reasoning in children (Morrison et al., 2006) WM model for multi-modal dual task (motor and verbal) (Morrison et al., 2001) Analogical performance in children with different cultural backgrounds (Doumas et al., 2010)
SPA	Serial memory task within a large-scale neural model (Eliasmith et al., 2012)

capacity at approximately 200 MB (Landauer, 1986), however, it is likely many orders of magnitude higher, as each of the trillions of synapses could potentially store 4.7 bits of information (Bartol Jr et al., 2015). For comparison, WordNet, the largest lexical database of words and semantic relations, contains 155,000 lemmas and requires 12 megabytes of storage in compressed form.[2]

Only several cognitive architectures have implemented large-scale memory. For instance, CHREST reported an LTM of 300,000 chunks representing chess patterns. From this large database, CHREST was able to retrieve specific patterns in under 250 ms, matching human expert performance (Lane et al., 2008). The narrow application domain of chess and similar board games likely contributed to these results by making memory contents more homogeneous and amenable to heuristics.

Another large model, TacAir-Soar, designed for a more complex aircraft piloting task, included 5,200 rules, 450 operators, and additional 575 MB of memory for storing the information about the environment and other agents (Laird and Jones, 1999). Despite the seemingly small number of items stored in the memory, significant optimizations were necessary to reduce latency and computational costs to enable real-time concurrent operation of multiple agents. These modifications included rewriting Soar in C from Lisp and enhancing the rule-matching algorithm.

Existing large-scale knowledge bases like WordNet and Cyc have also been explored as candidates for LTM but likewise required optimizations for fast and accurate retrieval of items. Here, the algorithms devised for database management and search have proved useful; Soar and ACT-R have been augmented with open-source relational database software to load concepts and relations from WordNet (Douglass et al., 2009). SPA holds the record for

[2] `https://wordnet.princeton.edu/documentation/20-`

the largest biologically plausible model of associative memory, capable of representing over 100,000 concepts in WordNet using a network of spiking neurons (Crawford et al., 2016).

Following the recent proliferation of large language models, their future potential as LTM components in cognitive architectures have been explored in several proposals (Choi, 2023; Joshi and Ustun, 2023; Romero et al., 2023). At the time of writing this chapter, there are no concrete results to discuss, however, numerous concerns have already been raised in other contexts regarding biological plausibility, reliability, and capabilities of generative models (see Chapter 11).

5.2 Memory types by contents

That memory is not unitary and consists of multiple systems with different properties has been established since at least the late 19th century (Squire, 2004). The dominant view in the early literature in philosophy (Ryle, 1945), psychology (Bruner, 1969), and artificial intelligence (AI) (Winograd, 1975) was that there is a dichotomy between "knowing what" and "knowing how." Experimental research in animals and patients with amnesia in 1960–1970s deepened our understanding of different types of memory and learning, as well as their physiological and neurological foundations. Building upon these findings, Squire and Zola (1988) formulated a hierarchical organization of memory, dividing it into two broad categories—declarative and non-declarative—each with several subgroups.

5.2.1 Declarative vs. non-declarative memory

Declarative and non-declarative memories are separated based on contents and properties. Declarative memory, also referred to as explicit memory, manages information about facts and events that can be recalled and "declared" to others. Non-declarative, or implicit, memory stores procedural knowledge (skills) that can only be expressed by demonstration, such as riding a bicycle.

Although nearly all cognitive architectures recognize the distinction between declarative and non-declarative memory, implementations of memory systems are very diverse. The choice of representation for memory is key since it determines what information is available to the system, how it is organized, and how it can be used. This, in turn, has effects on the design of the architecture and its performance.

Common knowledge representations

Developing knowledge representations remains one of the fundamental problems of AI and cognitive science. Davis et al. (1993) list the following five distinct roles of knowledge representations: 1) being a *surrogate* or a substitute for the things they represent, 2) an ontological *commitment*, i.e. to define what exists and what does not exist in the domain, 3) a fragmentary *theory of intelligent reasoning*, 4) a medium for *efficient computation*, and 5) a language for *expressing thoughts*. Because some of these requirements compete, choosing a representation for a cognitive architecture requires making intelligent tradeoffs. For example, a representation that describes the world in minute detail

might be complete and expressive but would likely complicate reasoning and increase computational demands.

Over the years, numerous knowledge representations have been developed with different properties. Cercone and McCalla (2012) list the following (mostly) symbolic representations: logical propositions, semantic networks, procedural representations, logic programming formalisms, frame-based representations, production system architectures, and knowledge representation languages. Shapiro (2006) distinguishes between logical and procedural representations, production systems, semantic networks, schemas (frames and scripts), pictorial representations, and connectionist representations (local and distributed). Sowa (1999) also considers logic, frames, rules, objects, and natural language representations.

A more recent taxonomy by Scharei et al. (2020) splits representations into symbolic, statistical, and subsymbolic types. In the symbolic category, there are two subgroups: logic-based (e.g. predicate, propositional, or other kinds of logics) and knowledge-based (e.g. ontologies and databases). There is some overlap between subsymbolic representations, such as neural networks, and statistical representations, which include probabilistic methods and machine learning approaches (e.g. Bayesian networks and Markov models).

Using these categorizations as a guide, below we describe five different types of representations found in cognitive architectures.

Logic formalisms. Logical expressions were historically one of the earliest forms of explicit knowledge representations and are a natural fit for expressing statements about entities and their properties. Furthermore, as Mylopoulos (1980) notes, the availability of inference rules, useful for theorem-proving and problem-solving, formal semantics (at least for the first-order and propositional logic), and simplicity of notation, make it relatively easy to implement and understand knowledge represented in such a fashion. Since this formulation is more suitable for expressing facts and beliefs, it is predominantly used for declarative knowledge. For example, predicate logic is used to express beliefs and goals in PRODIGY (Blythe et al., 1992) and PRS (Ljungberg and Lucas, 1992), and to describe the current world state in MIDCA (Paisner et al., 2014).

Graph-based representations. This group includes representations that have a network-like structure, i.e. those consisting of a set of nodes and connections between them.

A typical example of graph-based representation is a semantic network. The concept was proposed by Quillian (1967) as a psychologically motivated model of declarative memory, where nodes in the network correspond to concepts and edges between the nodes represent relationships between the concepts (e.g. conjunctive, disjunctive, subordinate, superordinate, modifier). This organization of knowledge is very expressive and by design captures the meanings of concepts through their connections to other concepts. In fact, this property is the basis of spreading-activation theory of semantic processing, sentence comprehension, and categorization (Colins and Loftus, 1975). Semantic search and priming are central properties of memory and are enabled by the same process. The memory search starts by selecting nodes in the network corresponding to the input concepts, and then tracing the links from each initial node in parallel until an intersection is found. The traced path which matches the syntax of the input best is selected. A similar process enables priming—

another important property of memory by which experience can influence current behavior. To prime a concept, activation from the corresponding node is spread throughout the network, proportional to the distance traversed. Then, when a new input is given, the intersection with the primed concept can affect processing proportional to the degree of activation. In other words, closely related concepts (e.g. "wings" and "bird") would be processed quicker than unrelated ones (e.g. "book" and "bird").

Graph-based and logic-based representations have equivalent expressive power, the differences between them being largely notational (Sowa, 2006). However, as Cercone and McCalla (2012) note, these notational differences are still important. For example, associative matching and intersection search operations are natural consequences of the graph-based notation but would likely not have been discovered if the logic expressions had been used instead.

Semantic networks are a natural choice for declarative and working memory, and are used in this capacity in many cognitive architectures, such as ACT-R, AIS, Clarion, Copycat, LISA, Soar, and SHRUTI.

Frames were proposed by Minsky (1974) as an alternative to logic- and graph-based representations. Frames are data structures that represent various concepts and phenomena with attributes (called slots) that can be assigned values, such as identifiers and other frames, or left empty to indicate uncertainty. From this point of view, frames may appear as yet another alternative to predicate logic, much like semantic networks. Below, we briefly discuss how frames differ from both, starting with logic.

Hayes (1981) discusses three inference types supported by frames: instantiation, criteriality, and matching. Instantiation, or creating instances of the same concepts, is naturally supported by frames. All that is needed is to create multiple frames with different values assigned to slots. Criteriality appears in perceptual reasoning in the following way: if an object, say a letter, is represented by a frame with slots corresponding to strokes that make up that letter, then the discovery of all the parts and filling the corresponding slots means that the letter is being perceived. Matching involves finding whether an instance of one concept (e.g. a man named Bob) can be considered an instance of another concept (e.g. a dog-owner), which naturally requires the context in which the frames are defined. All three inference types have (less intuitive) equivalents in predicate logic. What Hayes (1981) sees as the central limitation of frames is the difficulty in treating an instance of one concept as if it were another to discover their similarities and differences, which is useful for analogical reasoning.[3] The inherent ambiguity of frames is difficult to express in predicate logic, as it forces the choice of a single interpretation. Another property of frames that is difficult to express logically is the default values of slots.

Compared to graph-based representations, frames are similar in that they have nodes and relations. The difference lies in what these nodes and relations can represent (Alfonseca, 1989). Nodes in semantic networks are less expressive and only hold their labels, whereas frames are not limited in the amount and the type of information that can be placed in slots. Relations, on the other

[3]It is not impossible, however, as Tsotsos et al. (1980) used similarity links to relate frames describing different types of heart motion.

hand, are more restrictive in frames, where a single class of relations (often "is-a") enables the inheritance of properties from one node to another. In semantic networks, any type of relation is allowed. In addition, frames are related to object-oriented representations used in many modern programming languages, since they allow inheritance, definition of types, and creation of instances (Lassila, 1990). Objects in these languages, however, are more restrictive than frames in what inheritance relations and basic types they allow.

Frames were designed for representing declarative knowledge and are used for this purpose in several cognitive architectures, including FORR, ARCADIA, Disciple, COGNET, OMAR, and MIDAS.

Rule-based representations are simply condition-action pairs, where the antecedent describes the necessary condition for the action in the consequent component. Naturally, rules best accommodate procedural knowledge or what to do. Rules originated from and are predominantly known as part of the production system design (described in Section 2.2) and later expert systems (Hayes-Roth, 1985b). In both, rules are combined with the knowledge base (e.g. a semantic network) and an interpreter which searches for the rules whose condition matches the symbols in the database and executes the corresponding actions. Many of the well-known cognitive architectures, such as ACT-R, Soar, MHP, EPIC, CAPS, CHREST, BB1, and RCS, implement procedural knowledge as production rules.

Artificial neural networks (ANNs) have a long history and appeared under a host of different names, such as connectionist models, perceptrons, and parallel distributed processors. An ANN is a weighted graph, where nodes are identical simple processing units, each with an associated activation function, bias, and threshold. The main distinguishing feature of this representation is its distributed nature, meaning that groups of nodes rather than single nodes in the network correspond to individual items of knowledge, be it a concept or an action.

ANNs are used in a number of cognitive architectures: in SASE and MDB for encoding declarative memory and in DAC, HCA, and MBCA as a sensorimotor representation that combines both declarative and procedural knowledge.

Biological neural networks (BNNs) are close relatives of ANNs. BNNs are typically distinguished by their use of biologically motivated neuron models (e.g. spiking neurons) and learning rules (e.g. Hebbian learning). However, the line between ANNs and BNNs is blurry, since engineered and biologically motivated elements can be combined in a single network. Examples of architectures that use more biologically motivated neural representations for declarative and procedural knowledge are ART, BBD, Leabra, and SPA.

A large number of architectures use representations that cannot be categorized as any of the above type but combine some of their features. One of the common designs for procedural knowledge is a hierarchical network that organizes rule-based expressions or frames representing atomic actions into complex plans for a given task or series of motor commands. Many possible variations of this representation have been proposed: Reactive Action Packages or RAPs (Firby, 1989) used in MIDAS and 3T architectures, activity hierarchies in CARACaS (Huntsberger and Stoica, 2010), networks of tasks

in IMPRINT (Wojciechowski, 2004), action sequencer in ATLANTIS (Gat, 1998), and plan nets in ERE (Bresina and Drummond, 1990), to name a few.

Notable memory systems

Memory systems predicated on the separation of declarative and procedural knowledge are the most common and developed memory implementations. ACT-R implements an archetypal example of such a memory system, which influenced designs of many cognitive architectures.

In ACT-R, items in the declarative memory (called chunks) represent factual information organized in a semantic network, where each chunk corresponds to an individual node and edges reflect connections between related concepts. Procedural memory stores production rules in the form of if-then statements that specify a condition for the application of the action and an action to perform if the condition is satisfied. When a production rule is invoked, a relevant chunk is retrieved from declarative memory.

An important property of the semantic network in ACT-R is the concept of activation—a real number associated with each chunk that indicates how frequently it has been in use. Activation can spread along the links from the goal to the declarative memory elements linked to it. Continuous activation is considered a subsymbolic component that greatly expands the representational power of otherwise purely symbolic semantic network and allows modeling various WM and associative memory phenomena (Anderson, 1983b).

This memory implementation was very successful and provided explanations for many psychological phenomena, as well as hypotheses for further investigations. As a result, a number of cognitive architectures follow a similar design with some variations:

EPIC supports distinct declarative and procedural LTM storing propositions and production rules, respectively. Unlike ACT-R, where WM is a subset of LTM containing items with activations above a set threshold, EPIC's WM is structurally separate (Meyer and Kieras, 1997).

CAPS also maintains explicit declarative/procedural and LTM/WM separation. LTM holds classes (templates) for known objects and facts, and only their instances reside in WM. Activation can spread positively, increasing the activations of neighboring items, or negatively, inhibiting them (Sanner, 1999).

CHREST does not explicitly separate declarative and procedural knowledge; both are stored in a single data structure called the hierarchical discrimination network (Lloyd-Kelly et al., 2014). The network's nodes correspond to chunks (declarative knowledge) connected by links, either similarity or production, that provide matching and retrieval functions, respectively. Furthermore, this network is purely symbolic, with no numeric activations assigned to nodes. Instead, the network's topological structure represents a statistical distribution of known items, since the order and frequency of presentation of external information affects how the nodes and links are formed. The nodes are tagged with modality (action, auditory, or visual) and are self-contained, i.e. all information stored in the node is copied into it, which provides a speed advantage. In ACT-R, for comparison, each chunk has a single location in the LTM, which is referenced, not copied (Lloyd-Kelly et al., 2015).

Soar stores all procedural, declarative, and episodic knowledge in a single production memory. Different from the systems above, Soar's production rules

perform only memory retrievals, not actions or logical implications. A procedural/declarative distinction still exists in how knowledge is encoded: procedurally encoded rules retrieve only the structures encoded in the "then" part of the rule, whereas declaratively encoded rules retrieve all items. Thus, production rules perform double duty as conditional information or an indexed storage of declarative structures (Laird, 2022b).

Clarion supports separation of declarative and procedural knowledge, as well as a distinction between explicit and implicit knowledge. The implementation of the first distinction follows other architectures in the group: declarative knowledge (called non-action-centered) is stored in a semantic network with the nodes corresponding to chunks and associative links between them, and procedural (action-centered) knowledge is represented as action rules. However, explicit and implicit knowledge are also separated on the structural and representational levels, unlike ACT-R in which every chunk can be seen as a hybrid of explicit symbolic (labels) and implicit subsymbolic (activations) representations. Separation along both axes offers more flexibility in what level of accessibility to assign to each item of knowledge, whether procedural or declarative. Thus, every memory module in Clarion has a top and a bottom level for explicit (symbolic localist) and implicit (distributed) representations, respectively. Specifically, declarative chunks have symbolic nodes on the top level and dimension-value pairs organized in an associative neural network at the bottom level. Similarly, procedural knowledge is represented with rules on the top and neural networks at the bottom. Operations and learning can happen on either or both levels. To ensure consistency and coordination, implicit and explicit representations of concepts are linked together.

5.2.2 Semantic vs. episodic memory

The division of declarative memory into semantic memory for storing generic facts and episodic memory for autobiographical knowledge was proposed by (Tulving, 1972). Originally intended as a taxonomical distinction, it was controversial at first, but gained more acceptance in the psychological literature later (Tulving et al., 2002). Further empirical evidence showed that semantic and episodic memory systems in the brain are functionally and neurally disassociable but tightly integrated at the same time (Renoult et al., 2019).

The main feature of episodic memory is that it is personally experienced and tied to a specific location and time. For example, the names of the planets in the Solar System would be stored in semantic memory, whereas a personal experience of observing a planet through the telescope is part of the episodic memory. In an artificial system, a similar distinction can be made between generic knowledge and stored records of the past experiences. Below, we summarize main characteristics of episodic memory in cognitive architectures.

The purpose of episodes. Human episodic memory performs multiple functions, such as keeping a record of progress toward goals, preserving temporal properties of past events, and providing a way of forming concepts from experience (Conway, 2008).

In cognitive architectures, the main purpose of episodic memory is preservation of a chronologically ordered record of past events mainly for logging

and debugging purposes. For instance, in ACT-R, an episodic memory system helps in recovery from interruptions and ignoring irrelevant goals (Trafton et al., 2013). CogPrime uses episodic memory to learn new concepts from past events it experienced or hypothesized about (Goertzel and Duong, 2009), while Soar, ICARUS, and IMA use episodic memory to improve decision-making by reusing memories in familiar situations or using them as a guide in unfamiliar ones.

Contents of episodes. Time and context are two defining features of episodic memory. The simplest way to encode time is by associating episodes with timestamps (as in Clarion, GLAIR, and Sigma) or by storing them in the order of their arrival instead (as in MDB and CHREST).

What constitutes context varies among cognitive architectures, but typically includes a snapshot of any sensory inputs, memory state, action, and its outcome. This approach was adopted in Clarion, CogPrime, Soar, and CHREST. LIDA and IMA also store emotions associated with the event in episodic memory (Ramamurthy et al., 2006; Kawamura et al., 2008).

Storage and retrieval. Deciding what needs to be remembered and what does not is not trivial. The approach taken in many cognitive architectures, such as ACT-R, CHREST, MDB, and Soar, is to trigger the creation of a new episode when a new action is taken or a goal is achieved. However, there is a possibility of missing significant events during which the agent had remained inactive. To solve this issue, ICARUS triggers episode encoding when a rare event, such as a rarely seen predicate or a missing concept, occurs (Menager and Choi, 2016).

Memory should also accommodate an efficient retrieval mechanism. For example, Soar can retrieve an episode even from a partial cue (Nuxoll and Laird, 2007). However, search may become expensive if computational resources are limited and inaccurate as more episodes are accumulated in the memory. A solution to this problem is to periodically organize and compact entries in the episodic memory. In ICARUS and IMA, episodes are clustered for easier reference (Kawamura et al., 2008; Menager and Choi, 2016). CogPrime implements an attractor network trained on episodes, which serves as an index for retrieval and can also generate new episodes based on cues (Goertzel and Duong, 2009).

Episodic/semantic distinction is not as common as semantic/procedural in cognitive architectures; only a third of the projects in our selection contain episodic memory. Compared to other memory systems, the overall treatment of episodic memory is superficial and implementations are rudimentary. Furthermore, descriptions of the implemented memory systems often lack sufficient details to determine their structure and function.

5.3 Forgetting

Most discussions about memory focus on information retention rather than loss. Intuitively, if the main function of memory is preserving stored information, then forgetting should not be part of it. However, recent research suggests that forgetting is an essential aspect of memory function that actively prunes memory to improve retrieval and learning (Gravitz, 2019). Implementations of

perfect memory in computational models bring about a host of issues, further supporting the idea that forgetting is a feature of human memory rather than a bug.

Why forget at all? As a biological or artificial organism performs tasks and interacts with the world, it accumulates knowledge, but because the memory size grows, it takes longer and longer to search the memory for the necessary information. This phenomenon is endemic to many symbolic systems and is known as the *utility problem* (Minton, 1990). For example, Soar and ACT-R cannot learn continuously. When both systems were run on simple block stacking problems, their performance degraded and eventually failed after solving only a few hundred instances (Kennedy and Trafton, 2007).

The usual solution is forgetting some learned information, for example, using various heuristics, based on recency, utility, or even by discarding randomly selected memory items (Markovitch and Scott, 1988). The benefits of forgetting are not limited to the symbolic architectures. In the connectionist systems (e.g. DAC, BBD, and SASE), forgetting controls how soon and how well new information can be learned, improving the adaptability and generalizability to new situations.

How to forget? Save for brain injury, people rarely forget something entirely and all at once. Under normal circumstances, forgetting is well characterized by a power law, i.e. initially the rate of information decay is fast but it slows down as time passes (Rubin and Wenzel, 1996).

A number of power and exponential functions have been fitted to human and animal data (White, 2001), and have been used in cognitive architectures to imitate the gradual process of forgetting. A decay function expressed as t^{-d}, where t is the age of the memory item and d is the decay rate, was first used in ACT-R (Anderson and Schooler, 1991) and is generally supported by the psychological data. Other similar architectures that assign continuous values to items in the memory, such as Clarion, BECCA, ICARUS, FORR, IMA, NARS, Soar, SAL, SASE, and DUAL, also adopted the ACT-R-like equation with some modifications and different constants.

Other forgetting mechanisms are possible, but less common. For example, CAPS does not have a decay function, thus forgetting emerges as a function of the changes of activation spreading and capacity limitations (Thibadeau et al., 1982). In BBD and ART modifiable weights uniformly decay back to their original values (Krichmar, 2000; Grossberg and Seidman, 2006).

What to forget? People can forget both facts they once knew and skills they possessed. Those cognitive architectures that make distinction between different types of knowledge explicitly, generally model the loss of either declarative or procedural memory, but not both. Which memory is lost depends on the dominant representation and whether different types of knowledge are distinguished at all. ACT-R, CORTEX, FORR, PRODIGY, and LISA among others model decay of declarative memory, whereas Soar, BECCA, BBD, RCS, DAC, SASE, and IMA can forget procedural knowledge. Forgetting both is rare but can be implemented, as shown by Kim et al. (2007) who extended ACT-R with procedural skill degradation.

Is forgetting permanent? There are two options of dealing with forgotten items of memory. The first is to make forgotten items not accessible (e.g. due

to their low activation) but keep them in memory. This approach is taken in ACT-R, BECCA, ICARUS, and IMA. The second is to delete forgotten items, making them non-recoverable, as done in CORTEX, ERE, Soar, MicroPsi, LIDA, and PRODIGY. FORR and NARS take a middle ground: the items in memory remain available unless the storage space runs out, in which case the least used items are purged.

The first approach of retaining unused memory items is likely more cognitively plausible and helps avoid potential undesirable effects of permanently erasing memory items from storage. For example, if the knowledge in question is relatively encapsulated, it can be removed easily. For example, a salesman robot controlled by CORTEX can forget specific persons with all their associated details (Manso et al., 2014). However, if the removed items are connected to others, for example, in a commonly used graph-based knowledge representation, memory structure itself may become compromised. MicroPsi proposes to solve this issue by removing the items that became isolated and attempting to repair any gaps in the semantic network resulting from the degraded links to their neighbors (Bach, 2009).

5.4 Summary

- Because memory is essential for nearly all cognitive functions, it is present in virtually all cognitive architectures. For the most part, memory implementations conform to psychological evidence and distinguish between the major types of memory by persistence (ultra-short-term, short-term, and long-term) and contents (declarative, non-declarative).
- Working memory (WM) is the most complex and important part of the memory system, as it bridges perception with decision-making and long-term memory (LTM). Although most architectures implement an elaborate WM, often inspired by theories of human WM, few can demonstrate the associated behavioral phenomena or model processing at the level of neurons.
- Cognitive architectures use a variety of representations to store knowledge. Logic-, frame-, and graph-based structures are the most natural (and thus common) representations for declarative knowledge but can also be used for procedural knowledge. Rules are almost exclusively used to represent procedural knowledge, and network-based (localist or distributed) representations are equally suited for both declarative and procedural knowledge.
- Some aspects of memory, such as forgetting, and scaling to human-like capacity, as well as sensory and episodic memory, are not as well studied and deserve more attention.

6 Learning

Memory and learning are two closely related processes that are essential to human intelligent behavior. Memory allows storage and retrieval of information, experiences, and skills, while learning enables acquisition of new knowledge, adaptation to new situations, and modification of behavior based on past experience. Without a record in the memory, every experience and every thought would be like new, leaving no trace behind. Likewise, without learning, one is bound to repeat the past. The interplay between memory and learning is ubiquitous, continuous, and often effortless. In this chapter, we will explore various aspects of learning in cognitive architectures.

Section 6.1 examines the definitions of learning in psychology and artificial intelligence (AI).
Section 6.2 describes various taxonomies of learning.
Sections 6.3 and 6.4 investigate declarative and non-declarative learning in cognitive architectures, as well as learning strategies and the types of learning involved.
Section 6.5 discusses initial conditions needed for successful learning.
Section 6.6 covers cognitive architectures that do not implement any learning.

6.1 What is learning?

Before discussing the types of learning in cognitive architectures, we need to define the term "learning" itself. Like other familiar concepts, such as intelligence, attention, and autonomy, its meaning appears obvious at first glance but proves elusive on closer examination and differs depending on the discipline involved. In this section, we will present psychological and AI perspectives on learning, both of which are relevant for cognitive architectures.

6.1.1 Learning in psychology

Learning has been one of the core research topics in psychology for over a century, yet the concept of learning still has no precise and universally agreed-upon definition. Colloquially, learning refers to improvement of skill or knowledge, usually through trial and error. This is seconded in a more formal definition of learning as "a relatively permanent change in behavior potentiality which occurs as a result of reinforced practice" given by Hilgard and Marquis (1940, p. 6). Upon closer inspection, this definition is not complete and next we will examine its components to see what is missing.

Necessity of change. Learning itself is not directly observable, therefore changes in behavior are typically used to identify whether learning has occurred. However, as noted by Lachman (1997)

the relationship between behavior and learning is not one-to-one. In other words, a) learning is not the only mechanism capable of affecting behavior and b) lack of changes in behavior does not imply lack of learning.

To the first point, most biological organisms go through phases of development and aging in their life cycle. They can also experience a temporary or permanent loss of ability. Both of these factors can lead to observable changes in behaviors but on their own are not considered learning, even though some adaptation likely occurs as a result, e.g. the maturing and decline of visual ability as reviewed in (Siu and Murphy, 2018).

Examples of situations when learning took place, but behavior has not changed are classical conditioning and latent behaviors. In the case of conditioning, learning affects the stimuli effectiveness but not the behavior itself (Lachman, 1997), whereas latent behavior changes may be revealed long after learning has occurred. For example, an athlete may learn a new strategy by observing others but will not be able to apply the new knowledge until the next competition (Olson and Hergenhahn, 2016, p. 4).

In sum, changes in behavior alone are neither necessary nor sufficient to conclude that learning has taken place.

Relating change to behavior. De Houwer et al. (2013) note that detecting changes in the organism and relating them to preceding experiences is very difficult for two reasons. First, the mechanisms involved in learning are not known in advance, making their detection problematic. Second, distinguishing whether experiences and changes in behaviors are linked causally or merely correlated may not always be possible.

Permanence of change. For a change in the observed behavior to be considered a result of learning, it should be relatively stable. This delineates learning from the transient changes in behavior due to variations in motivation, sensory adaptation, fatigue, or effects of substances (Houston, 1981; Lachman, 1997).

Necessity of deliberate practice. Many types of learning are deliberate and repetitive, such as training sports skills or mastering a musical instrument. However, learning may also happen incidentally or even as a result of a single event. As an example of the latter, observing others perform a task differently may be sufficient for adopting it in the future (Lachman, 1997). It is also possible that most of the learning happens unintentionally because learners often have no recollection of when and how it occurred (Alexander et al., 2009). Such learning, however, can be easily conflated with unpracticed cognitive skills that result from maturation or aging (Houston, 1981).

Not all of these concerns apply to the types of learning implemented in the existing cognitive architectures. For example, the vast majority of cognitive architectures do not model development, aging, or temporary loss of ability. Even when these issues are considered, the lifespans of the artificial agents are too short for significant changes to occur.

6.1.2 Learning in AI

In the artificial intelligence domain, learning is primarily investigated within the machine learning (ML) paradigm. ML aims to acquire knowledge by extracting patterns from data, using techniques from various disciplines to modify or adapt the output toward a desired outcome. The necessary components

of learning can thus be defined as 1) a learning objective, 2) training data, 3) training procedure, and 4) measure of performance. Learning takes place if performance measures of the algorithm are improved as a result of training on particular data for the given task.

There are some parallels between the psychological definition of learning and ML: in both there is an observable change in the behavior of the artificial system as well as practice and reinforcement signal (e.g. provided as ground truth or obtained by the system through interaction with the environment). However, there are some differences too. ML is investigated primarily on the algorithmic level, which focuses on the complexity of the proposed methods and their effectiveness for different applications (Langley, 1996b). Although some ML methods may be cognitively or biologically plausible (whether intentionally or not), the bulk of ML research is not concerned with modeling or explaining the mechanisms of human learning. According to Mitchell (2006), ML is driven primarily by statistics and computer science and less so by studies of human learning, partly due to poor understanding of the processes involved.

6.2 Types of learning

Learning is a complex process that can occur in many ways. To categorize types of learning and their properties, a number of taxonomies have been proposed. According to Kyllonen and Shute (1988), the most common divisions are by outcomes of learning and types of information processing involved. Both will be explained below:

Designated/rational. Following behaviorist tradition, it is possible to categorize learning solely by its outcomes. Two well-known categorizations of this type are Bloom's (1956) taxonomy of educational objectives and Melton's (1964) taxonomy of learning. Bloom organizes educational behaviors into six classes from the simpler to more complex, such as knowledge, comprehension, application, analysis, synthesis, and evaluation. Along the same lines, Melton ranks the following learning acts in order of increasing difficulty: conditioning, rote learning (memorization), probability learning, skill learning, concept learning, and problem-solving.

Information processing. These are rooted in theories of information processing and implementations in computational models. The taxonomy based on ACT-R (Anderson, 1983b) divides learning into three basic types: declarative learning, procedural learning, and strengthening, similar to reinforcement, that improves the other two. More general and somewhat overlapping taxonomies that address information processing were proposed by Carbonell et al. (1983) and Michalski (1986). Carbonell et al.'s (1983) taxonomy comprises learning strategies (ordered by complexity from memorization to unsupervised discovery), representation (e.g. parameters, decision trees, logic-based expressions, networks), and application domains (e.g. general methods, playing games, mathematics, natural language processing). Michalski (1986), in addition to the learning strategies, also considers the dimensions of research paradigm (connectionism, symbolism, and domain-specific learning) and learning orientation (general learning algorithms, models of human learning, specialized engineering solutions).

In AI, the most common categorization is based on learning algorithms divided by the type of training experience into supervised, unsupervised, and reinforcement learning:

Supervised learning requires a set of inputs with target values. The algorithm learns to map inputs to desired outputs (discrete categories or continuous values). To be of use, the algorithm needs to generalize past training inputs and produce reasonable outputs for unseen data. This is what is known as *inductive learning*. Besides inductive, there are transductive learning algorithms, which reason from specific training cases to specific test cases.

Unsupervised learning, as the name suggests, is not provided with target values, only with the inputs. During training, the algorithm is expected to determine the distribution of data or cluster it into discrete categories.

Self-supervised learning is a form of supervised learning that does not require extensive manual labeling. Even though it learns from unlabeled data, it does not perform clustering or grouping as the unsupervised approaches do. Instead, it exploits the unlabeled data to yield labels, for example, removing words/pixels from sentences/images and learning to predict the missing parts. This type of learning can be useful on its own or as a way to form meaningful representations for other tasks that may or may not require supervision to learn.

Semi-supervised learning requires only a small set of labeled data, which is then used to bootstrap learning on a much larger set of unlabeled data.

Reinforcement learning is sometimes included in the unsupervised category (Barber, 2012), but more often it is considered as another approach having features of both supervised and unsupervised learning. In the reinforcement learning paradigm, the algorithm is not told what to do and is not provided with target values explicitly but can perform actions and assess their effects through interactions with the environment.

Meta-learning generalizes to multiple training tasks instead of individual training samples to form inductive biases based on the previous training experience.

There is no universally accepted taxonomy of learning methods in cognitive architectures, however discussions of learning naturally fit within information-processing taxonomies. The majority of works categorize learning by the memory system that supports it, typically following Squire's (1992) hierarchical taxonomy (see Chapter 5). For the remainder of this chapter, we will use the memory-based taxonomy as well. In the discussions of declarative and procedural learning, approaches are further divided by learning strategies and learning algorithms.

6.3 Declarative learning

Declarative learning extends declarative knowledge about the world, e.g. objects, their properties, relationships between them, physical laws that govern their behavior, etc., expressed as facts, theories, or constraints. Knowledge can be extended in two ways: adding new information, such as learning about the

objects that were never encountered before, or modifying existing knowledge based on new evidence. Below, we discuss examples of declarative learning organized by learning strategy and level of supervision.

6.3.1 Learning through instruction

Learning through instruction is a way to add or update knowledge of the system without interrupting its operation and reprogramming directly. This process, however, requires a human teacher and some scaffolding on the side of cognitive architecture, such as a mechanism to detect when a teacher is instructing, means of recording and parsing the instruction, a way of transforming the information into suitable internal representation, and an initial knowledge base to process and integrate obtained knowledge.

Only a few limited examples of such learning exist. One is Rosie, an agent implemented in Soar that can learn new concepts, such as objects in the environment and their properties, via natural language instructions (Mohan et al., 2012). Rosie's background knowledge includes basic perceptual properties (color, shape, and size) and a way of extracting them from sensors and primitives for spatial relations (e.g. direction and alignment). With this in place, Rosie can learn novel concepts and relationships between them from a handful of examples given by the teacher. For instance, it can learn what "left of" means from a single example, such as "the red object is left of the blue object." In addition to passive learning of input information, it actively seeks input to resolve ambiguities by asking additional questions.

Along similar lines, ISAC learns names of objects by associating object classifier output, image histogram, and other properties with the provided label (Kawamura et al., 2008). Another cognitive architecture, PRODIGY, implements interaction via a graphical interface, through which it receives domain knowledge facts and instructions (Blythe et al., 1997). The interface also provides users with access to the internal representation and intermediate processing steps, which allows more fine-grained guidance during problem-solving.

6.3.2 Learning by deduction

Learning by deduction is the application of predetermined inference rules to existing knowledge. One of the main features of deduction is truth preservation, i.e. provided that the premises are true and inference is done correctly, the conclusion is also guaranteed to be true. For example, in the syllogism "All humans are mortal. Socrates is human. Therefore, Socrates is mortal," the truth of the final conclusion rests on the initial knowledge that humans are mortal and Socrates is a human.

Given any set of premises, deductive inference can be applied repeatedly without supervision (referred to as forward chaining) and stop when a deductive closure (all possible derivations) of the set is computed. This process is one of the main methods of extending knowledge in symbolic (or mostly symbolic) architectures, such as Clarion, Disciple, NARS, and OSCAR; deductive inference from a knowledge base generates new facts which are then saved in memory to guide and simplify future derivations. Deductive learning is not limited to symbolic approaches; it has been demonstrated by SHRUTI that

deductive inference can also be performed using the connectionist principles (Shastri, 1990).

Two issues are brought up frequently with respect to learning by deduction. First is that deduction is not learning since it does not add new knowledge but rather rearranges the information already contained in the premises. Thus the primary benefit is seen as simply making information more readily accessible. The second issue follows from the definition of deduction, whose truth value depends on the truth of the premises. Therefore, errors or missing information in the initial knowledge base may either lead to incorrect statements or make it impossible to infer correct statements.

6.3.3 Learning by induction

Induction helps derive general conclusions from a set of specific observations or given examples, a process opposite to deduction which follows from general axioms toward specific facts. For example, knowing that eagles and sparrows fly, it is reasonable to conclude that all birds can fly. Obviously, the existence of non-flying birds, penguins and ostriches, refutes the conclusion. Therefore, induction is not guaranteed to be true, since even correct premises provide only partial evidence. Moreover, unlike deductive arguments which are all equally strong, some non-deductive arguments may be stronger than others, depending on how well-justified the premises are.

Despite being inherently probabilistic, inductive learning is still very useful because it can expand knowledge from a very small number of observations to a potentially infinite set, but at the cost of increasing uncertainty. For example, if the concepts are organized hierarchically, their properties can be transferred inductively to concepts at the upper or lower levels. Two examples from Clarion summarize this: 1) transfer of properties from general to specific, e.g. if we know that a sparrow is an instance of a bird and birds fly, then we can assume that sparrows fly too; 2) transfer of properties from specific to general, e.g. if we know that cats jump and that cats are animals, then we can assume that all animals jump (Sun, 2003). The CHREST architecture applies a similar approach to learning game board patterns by the way of familiarization and discrimination, respectively. Familiarization adds missing parts to the general representation (image) from incomplete views of the board (patterns). Discrimination works the other way around by extending the patterns to better match the image (Lane and Gobet, 2012). Analogy can also be applied to concepts on the same level in the hierarchy. For example, Disciple transfers knowledge from a known entity S to a similar but less known entity T. If S and T are similar in some respects (e.g. have the same features) they can be similar in other respects too. If S has some feature that T does not, it is reasonable to assume that T has it too (Tecuci, 1995).

Furthermore, induction improves domain knowledge by filling the gaps and removing inconsistencies, issues that cannot be resolved deductively. This approach is taken in Disciple, where negative and positive examples of rule applications are used to refine declarative knowledge about the domain of operation (Tecuci, 1991). For example, given a domain, where *x IS-IN y* presumes that *y IS-A container* and the solution to *TAKE x* requires to *OPEN* the container. However, *sink*, *box*, and *cabinet* defined as containers in the domain knowledge base do not have a property *IS closed*. Thus, when

attempting to take an object from the sink, *OPEN* action cannot be applied, which generates a negative exception. To resolve the issue, Disciple examines features of the objects in positive examples (e.g. taking x from the *box* and *cabinet*) and identifies objects in negative exceptions that do not have those features (*sink*). If *closed* is identified as a property that has not been assigned to all containers consistently, it is added to the knowledge base. Similarly, Disciple refines declarative knowledge by finding generalities among the features of objects and separating them as higher level concepts in the hierarchy, e.g. noticing that some objects can be moved and some cannot will prompt the system to add *movable* as a new property.

In most cases, inductive learning does not require supervision, since the process relies on accumulated knowledge. A variety of algorithms have been used to implement inductive learning in cognitive architectures: including heuristics (Disciple, CHREST, Clarion), backpropagation (Clarion), Hebbian learning (SHRUTI), or a combination thereof (LISA, Clarion). Given the uncertainty of induction, correctness of generalizations needs to be verified (e.g. via human feedback in Disciple) but this is often not done.

6.4 Non-declarative learning

Non-declarative learning refers to the process of acquiring skills, habits, and conditioned responses. This type of learning is implicit and occurs through experience and practice, rather than memorization. In this section, we consider two types of non-declarative learning: perceptual and procedural.

6.4.1 Perceptual learning

Perceptual learning improves the organism's abilities to respond to its environment as a result of experience or practice (Hawkey et al., 2004; Kellman and Massey, 2013). Since perception is implicit, non-verbalizable, and a part of many activities, it may be difficult to distinguish perceptual from procedural learning. From the epistemological point of view, the distinction is that perceptual learning must improve perception, not just involve it. Some examples of perceptual learning include: recognizing the differences which could not be perceived before (differentiation), perceiving distinct properties as one (unitization), learning to attend (attentional weighting), and the ability to recognize specific stimuli (stimulus imprinting) (Connolly and Prettyman, 2024). The first ability, differentiation, is foundational since it bootstraps other types of learning. A living or artificial organism that knows how to tell objects and sounds apart can extend its perceptual abilities further, learn about the properties and relationships among the perceived entities, formulate its goals, and discover what actions will lead to achieving set goals while avoiding harm.

Off-line learning

Most cognitive architectures are not concerned with developing perceptual abilities *per se*, but rather need perception for reasoning and decision-making. For this purpose, the internal workings of a model do not matter as long as it provides the information needed. Thus, prelearned black box models that remain constant throughout operation are quite common, especially when the

environment and the task are reasonably well-defined and are not expected to change. In this case, a set of relevant entities can be determined and used as a training set for supervised inductive learning. Although the specific algorithms applied and what they are applied to may be different, there are a number of similarities across various cognitive architectures we considered:

- The dominant modality is visual. One of the few exceptions is SASE that learns a limited vocabulary of spoken words (Zhang et al., 2005).
- Perceptual models are trained to perform a wide variety of visual tasks from low-level vision to image classification, object detection, object recognition (including face recognition), segmentation, tracking, and action recognition. Note, however, that all cognitive architectures are specialized in one or two of these tasks and none have a general purpose visual system.
- Learning is mostly inductive and supervised, using manually labeled data.
- Learned models are optimized for the object sets and environments where the cognitive architectures operate (e.g. video game avatars or real objects in natural environments). The widespread use of special purpose features and algorithms (e.g. for face recognition) may not be effective for more general vision tasks (e.g. generic object recognition).
- The number of recognized entities and events is usually small. One of the largest models, Leabra, recognizes up to a hundred categories of objects, which is about an order of magnitude above the number of entities that an average cognitive architecture can handle.

Online learning

An off-line approach to perception has an obvious limitation—it will likely not work well in the settings whose properties and conditions of operation differ from the ones used during training. As a result, prelearned models would need to be modified for any additional tasks or object categories. An alternative is online learning, i.e. gradual adaptation to changing environmental conditions. While such continuous learning is more biologically plausible, it is a much more difficult problem to solve, thus fewer cognitive architectures attempt it.

- Continuous learning needs to be bootstrapped by initial knowledge. In some cases, the pretrained model can serve as a source, e.g. Kismet uses a face detector to learn features for further unsupervised person identification and relies on a speech recognizer for understanding the teacher's commands. In some cases, a set of simple heuristics is sufficient. For example, in RCS and Kismet, certain features are chosen to aid object recognition (Albus et al., 2006; Aryananda, 2001).

  Reflexive hard-wired responses can indirectly guide perceptual learning: in BBD, appetitive and aversive responses are triggered by taste (Krichmar and Edelman, 2002), DAC has approach and avoidance reflexes (Verschure and Voegtlin, 1998), and IMA is set up with commands that start learning and assign rewards (Kawamura et al., 2008).
- Continuous learning algorithms exist for any level of supervision. Human feedback is the most common. For example, IMA relies on a teacher to introduce objects and people, whose features are then saved in the database and retrieved when necessary (Kawamura et al., 1995). A robot controlled by Soar likewise follows human instructions. A spoken command "this is orange" prompts Soar to associate certain visual features with the label

"orange" (Mohan et al., 2012). DIARC similarly populates its database of objects through instruction. When a new object is encountered, the robot queries the human user, who may then provide a label (Scheutz et al., 2007).

Semi-supervised learning is demonstrated by RCS, which initially relies on radar to learn the differences in the appearance of drivable areas and obstacles in supervised fashion. However, due to the sensor noise, these labels may not be correct, therefore the movement of the vehicle and response of its bumper are used to determine traversability of the terrain and correct the labels if needed (Albus et al., 2006).

Unsupervised perceptual learning is done primarily via clustering. For example, Kismet passively observes the environment and gathers samples of the users it is interacting with. When a sufficient amount of data is accumulated, it is clustered and the centers of clusters are assigned unique identities. When a new person appears in front of the camera, their face is compared to the existing clusters. If no match is found, a new cluster is automatically generated (Aryananda, 2001).

- Vision is still the most common modality for continuous learning, with few exceptions. One is Kismet, which learns to understand speech directed at it. Starting with the prelearned speech recognizer and known prompts, it populates its vocabulary and then uses it to identify known utterances or add new ones to the vocabulary following the user's verbal instruction (Varchavskaia et al., 2001). Another notable example, BBD, learns to associate percepts from three distinct modalities—visual, auditory, and gustatory. One of the instances of BBD, Darwin VII, demonstrated how learning occurs through an unsupervised operant conditioning process. First, it learns to associate visual patterns (blobs, vertical and horizontal stripes) with an innate value-loaded test, and then pairs the visual patterns with the auditory tone emitted by the object (Krichmar and Edelman, 2005).

Overall, online perceptual learning has shown potential but does not reach the level of performance of the off-line models even in simple environments.

6.4.2 Procedural learning

In the most general sense, procedural knowledge describes *how* to do something. "Doing" usually refers to the strategies for solving problems and sequences of motor commands to accomplish the tasks, or a combination of both. Learning new cognitive and physical skills or improving existing ones is called procedural learning. In this section, we will discuss several learning strategies, specific algorithms used to implement them, and what level of supervision they require.

6.4.3 Learning through instruction

Earlier, we described examples of declarative learning through instruction. Along the same lines, human users can teach procedural knowledge directly (e.g. by demonstration) or indirectly (e.g. by providing feedback). Direct instruction is usually applied to learning motor commands, since they are easier to demonstrate. For example, IMA records motion exemplars for actions while the human user controls the robot via teleoperation and provides spoken

labels for the performed action (Kawamura et al., 2008). Similarly, association between servomotor commands and high-level instructions is learned in CogPrime (Goertzel et al., 2010a) and SASE (Zhang et al., 2005) when the user simultaneously moves the robot and feeds the command. Human expert supervision can also be applied to problem-solving episodes, as in the case of Disciple, which prompts the user with questions and can understand the explanations, examples, and hints provided via the graphical user interface (Tecuci et al., 2004).

6.4.4 Learning by deduction

The simplest way of learning procedural knowledge is caching the results of the successful executions. Whenever a task is completed, be it a solved problem or physical movement, a sequence of actions that led to the solution is recorded and reused later. Naturally, caching is most effective in the cases when the same problems are encountered frequently and generating a solution every time is computationally expensive. For example, the robotic architectures that use sophisticated planners to compute a route to the destination benefit from saving successfully traversed paths since there is no need to compute them again. In limited indoor spaces those same routes are likely to be reused multiple times, as demonstrated in the experiments using the BB1 architecture for controlling an office delivery robot (Hayes-Roth et al., 1993). Robots operating in larger outdoor environments can also benefit from this technique. For example, the RCS architecture caches successfully completed routes but before committing them to memory performs postprocessing to remove loops and excursions (Albus et al., 2006). Another robot architecture, CORTEX, further improves caching by saving only those plans that are sufficiently different from those already in memory using a meta-descriptor for comparisons (Manso et al., 2014).

A more sophisticated alternative to caching is explanation-based learning (EBL) (DeJong and Mooney, 1986; Mitchell et al., 1986). It is one of the most well-known deductive learning methods that gained popularity during the 1980s and was featured in several cognitive architectures developed at that time, notably PRODIGY and Disciple. Various implementations and variants of EBL have two things in common: 1) the ability to deductively generalize from very few (or even a single) example, and 2) reliance on an extensive domain knowledge to do so. For any given specific example, EBL first "explains" it via deduction from a preprogrammed logical domain theory and then generalizes the example to be reused in similar situations.

EBL-like learning appears in several other architectures under different names. In Soar it is called chunking and is described as the process of combining a sequence of operators into a single operator (referred to as a *macro-operator* or *chunk*), which has the same effect as the original sequence. Considering that the production systems rely on forward chaining for reasoning, arriving at the sequence of actions is similar to the derivation step in EBL (Rosenbloom and Aasman, 1990). The creation of a macro-operator in turn corresponds to the analysis and generalization of the explanation. Soar, implements chunking that is closer to the original EBL (Laird et al., 1991). In Soar, chunking helps avoid impasses arising when more than one operator is applicable to the problem. Chunking thus eliminates deliberation and making

the action more reflexive. Chunking can apply to multiple rules and takes effect immediately (does not need to be reinforced through the application) (Laird, 2022a). In ACT-R, a similar process is called production compilation and provides a way to combine two production rules that have been used in sequence into a single rule. Essentially, this creates a specialized routine that is more efficient on subsequent applications in similar scenarios (Anderson et al., 2003). The term "chunk" in ACT-R refers only to items in the declarative memory (Anderson and Lebiere, 2003).

ICARUS is similar to Soar and PRODIGY in that it applies EBL to optimizing performance, and in addition it can create disjunctive and recursive skill hierarchies (Langley et al., 2005a). For storing skill in memory, the assumption is that the acquired skill clauses that achieve the same goal have the same head (condition for execution).[1] Indexing in this manner allows ICARUS to store two or more clauses together, which in turn helps create skills that call themselves directly or through others. This is more flexible than chunking in ACT-R and Soar (Langley et al., 2009).

ERE further extends EBL through the mutual theory refinement (MTR) method, which combines learning from failure and explanation for refining approximate or incomplete domain theories. When ERE detects failure (expressed as the difference between the expected and observed state after the action), it collects evidence, such as existing and missing literals in the predicted and observed states and attempts to heuristically resolve the inconsistency by either removing or adding preconditions or modifying the outcomes of the rules (Kedar and McKusick, 1992).

Although deductive approaches are effective, they have limitations in practice. First, methods like EBL require a complete knowledge base, which is unrealistic for most practical domains. Second, the amount of acquired knowledge can grow rapidly even for small domains. As a result, the time to search for a suitable rule to apply can offset the optimizations from learning new rules. This is known as a utility problem (Minton, 1990). Third, ultimately, deductive methods in general and EBL in particular cannot go beyond the deductive closure of the initial knowledge base. Therefore, other learning methods, such as learning by instruction, inductive learning, and exploration are needed to compensate for these drawbacks. Virtually all cognitive architectures mentioned above do not rely on deductive learning alone.

6.4.5 Learning by induction

As explained earlier, induction generalizes a few isolated experiences to a potentially unbounded set of situations but does not guarantee that the solution is universally true. In the symbolic systems, the simplest form of inductive learning is replacing individual instances with variables. For example, the Disciple architecture can generalize a rule that "pushing a door will open it" by replacing *door* with x (Tecuci et al., 2005). The new rule will be true for doors and other things that can open, but not for all objects. Disciple and

[1] ICARUS uses Horn clause formalism for encoding facts and skills. Both types consist of a head and a body. A head specifies the name of the concept or skill and arguments, and a body includes a number of fields. The fields in concept clauses can specify associated percepts or relations to other concepts. Skill clauses include fields that specify the conditions for executing the skill, description of the skill itself, and effects of its application (Langley, 2006a).

other architectures following this approach, such as DUAL (Kokinov, 1994b) and NARS (Wang, 2012), can to some extent verify the plausibility of the generalization with ontology and other rules but ultimately require additional input to check the correctness of the hypothesis. This brings us to by far the most common type of learning found in cognitive architectures.

6.4.6 Associative and reinforcement learning

Perhaps, the most fundamental learning strategy is learning by doing, observing the effects of actions, and modifying behavior accordingly, which is found in many living organisms. In natural sciences, two dominant theories, classical and operant conditioning, explain associative and reward-based learning, respectively. Classical conditioning associates an existing stimulus (e.g. food) that elicits certain involuntary behavior (e.g. salivation) with a new neutral stimulus (e.g. bell). As a result of learning, the old behavior (salivation) becomes a response to the previously neutral stimulus (bell). Operant (or instrumental) conditioning changes voluntary behavior by rewarding or punishing it. A variety of algorithms that exploit these principles were developed within AI and computer science, notably reinforcement learning, comprising algorithms that learn how to act based on numerical rewards assigned to different actions and states.

Reinforcement-like mechanisms that approximate instrumental conditioning are very common in cognitive architectures. One of the early successful applications is ACT-R, where each unit of procedural knowledge (production rule) is assigned a value representing its utility. Initially, all weights are equal and are adjusted depending on whether the application of the rule was successful. If so, the value of the production (and other productions that led to the reward) increases, and decreases otherwise. Reinforcements (positive and negative) gradually shape the distribution of utilities across production rules and improve the chances that rules that were more useful in the past will be selected again among alternatives (Fu and Anderson, 2006). Soar initially relied only on symbolic learning (chunking) and did not adopt a similar reinforcement mechanism based on the utility of operators and productions until the ninth version of the architecture (Laird, 2012b).

The same procedure can be applied to virtually any procedural knowledge representation. FORR encodes procedural knowledge as an ensemble of weak learners specialized for certain tasks called Advisors, each with an associated weight. The advisors whose outputs lead to better decisions are given higher weights, and those that consistently underperform eventually are dropped (Epstein and Petrovic, 2008). LIDA encodes procedural knowledge in schemes (called behavior codelets) that have activations. If the action performed by a codelet is successful (i.e. matches the expectations), it is reinforced in the memory and more likely to be selected again (D'Mello et al., 2006). Clarion represents procedural knowledge with both symbolic (chunks) and subsymbolic (neural networks) representations that are linked together. Subsymbolic representation is learned via Q-learning and backpropagation. Symbolic rules are then constructed and refined using inductive or deductive methods based on the success ratios of various modifications (Sun and Fleischer, 2012). CARACaS also uses Q-learning, but reformulates it in process algebra terms to learn new actions from observations (Huntsberger, 2011b).

The cognitive architectures discussed below aim at modeling classical and operant conditioning explicitly for learning various tasks. CHREST integrates a combination of classical and operant conditioning with symbolic reasoning for a task of playing a blackjack card game (Schiller and Gobet, 2012). Classical conditioning associates emotional tags (sadness or joy) with various actions. Depending on the emotional tags, the next action is selected, its immediate outcomes are observed, and the model adjusts its expectations by changing its parameters accordingly. DAC, BBD, ART, SASE, HCA, and Leabra explore neural mechanisms of conditioning. DAC models simultaneous perceptual and behavioral learning via classical and operant conditioning implemented as predictive Hebbian learning (Duff et al., 2010). As a result, a robot controlled by the system learns to navigate in a previously unknown environment. A different solution for learning to navigate was developed for Darwin IX (an instance of BBD) using a delayed eligibility trace (McKinstry et al., 2008). SASE applies a Q-learning approach to learn composite actions composed of primitives through conditioning with the feedback from the human instructor (Zhang and Weng, 2002). Finally, the PVLV model (based on the Leabra architecture) improves the Rescorla-Wagner learning rule to model a variety of properties of classical and operant conditioning, such as blocking, overshadowing and summation, conditioned inhibition, second-order conditioning, and CS-US timing variability, all in a single model (O'Reilly et al., 2007).

6.5 What is needed for learning?

No architecture starts from a blank slate, but how much and what kind of information is initially given can differ significantly across the architectures. Some examples of commonly preloaded knowledge are listed below.

Perceptual knowledge. Proliferation of off-the-shelf open-source and commercial software that performs basic perceptual tasks made it easier to integrate perception with virtually any cognitive architectures.

- *Visual processing* is frequently outsourced to plug-in modules and includes saliency estimation, object detection, object recognition, and scene understanding (obstacle classification, SLAM, segmentation, and optical flow estimation). Many robotics architectures resort to this because they prioritize robustness and effectiveness of vision over cognitive plausibility (e.g. see ADAPT, CARACaS, RCS, Saphira, SS-RICS, and TCA), although some cognitive architectures resort to these methods as well (e.g. ARCADIA, DUAL, Sigma, and STAR).
- *Speech processing* is most common for audio modality, given its primary purpose of communication in the majority of the architectures that have it. Again, off-the-shelf software is more commonly found in the robotic architectures, where it serves to translate commands from the users, as in ADAPT, CORTEX, DIARC, and Ymir.

Declarative knowledge. Obtaining declarative knowledge is difficult even in the simple environments, therefore this information is often preloaded as:

- *Domain constraints*, i.e. what can and cannot be done in the given environment, including laws of physics or rules specific to the task or situation.

- *Objects*—entities known to the system, including their types, appearance, and relationships between them (e.g. as an ontology or hierarchy).
- *Objectives*—task description or specification of the desired state of the system.

Procedural knowledge. Arguably, procedural knowledge is learned more often than perceptual or declarative knowledge. It can also be predefined, e.g. in the form of rules (ACT-R, EPIC, Soar, and other production systems), finite state machines (CARACaS, STAR, Subsumption), and task networks (3T, ATLANTIS, TCA).

Overall, cognitive architectures that can be preloaded with information are either symbolic or have a significant symbolic component for knowledge representation. This initial knowledge makes it easier to bootstrap further learning, but even without it many cognitive architectures are able to perform complex intelligent tasks.

Connectionist architectures, instead of knowledge base injection, need to be provided with ample scaffolding that may not count as knowledge but makes its acquisition feasible. For instance, many connectionist cognitive architectures have fewer sensors with lower resolution, operate in simpler environments with fewer objects, and perform simpler tasks. The structure of the model (e.g. number and type of neurons and their organization in a neural network) and formulation of learning procedure (e.g. learning objective, parameters of learning) have a tremendous effect on the learning and its outcomes. Even preloading knowledge is not entirely out of question. Distributed representations make it difficult to specify concrete facts or skills for specific scenarios, but can instead be pretrained on generic tasks, such as classification and detection of objects and actions. Knowledge transfer by learning domain-invariant models from large amounts of data is a common technique in deep learning (Long et al., 2015) but is not frequently used in the cognitive architecture domain.

6.6 Non-learning cognitive architectures

Even though learning is considered one of the top requirements for achieving intelligence, about one-third of the architectures in our selection do not learn. The architectures that do not learn can be subdivided into two groups: those that intend to implement learning in the future and those that do not aim at modeling learning in principle.

Nearly all architectures in the latter group focus on human performance modeling for specific applications and tasks. Their main purpose is to optimize the design process for human-facing interfaces. Thus, they focus on preventing design-facilitated errors by generating quantitative predictions of human performance for a range of known cognitive parameters while accounting for individual variability. Some models of this kind, such as MIDAS, COGNET, MAMID, and EPIC, do not include learning at all, while others do not learn themselves but can model aspects of human adaptability and learning. For instance, Model Human Processor (MHP) (Card et al., 1986) predicts learning and task execution time for different types of users, whereas APEX (Freed and Remington, 1999) models human error by introducing a temporary bias parameter that controls whether the decision mechanisms will rely on default settings or on a more costly working memory retrieval.

Sometimes, lack of learning is intentional. For example, Copycat (Hofstadter and Mitchell, 1994) and its successor Metacat (Marshall, 2006) adjust to the context during analogy-making episodes but revert to the initial state once the context was removed and do not retain changes across runs. While temporary adaptation is sufficient for dealing with different types of problems, it is unclear how learning can improve analogy-making in this context.

There are no architectures that cannot accommodate learning in principle. Arguably, some connectionist architectures allow learning more easily, and most that do not are either symbolic or have a substantial symbolic component. As we discussed earlier, connectionist architectures cannot be easily preloaded with data making learning a necessity, whereas symbolic architectures by design are capable of producing intelligent behavior without any learning at all and can be interfaced with external models of learning if needed, e.g. EPIC with the integrated model of human category learning (Wray and Chong, 2003).

6.7 Summary

- Learning is seen by many as a fundamental characteristic of intelligence. Psychology and cognitive science consider learning as a beneficial behavioral change resulting from experience but not from transient states, maturation, or chronic diseases. The AI perspective focuses mainly on developing algorithms for inductive learning and learning from experience with various levels of supervision.
- The importance of learning is recognized in cognitive architectures as well. Two-thirds of the projects we considered have at least a rudimentary ability to learn. Of the remaining architectures that do not learn, only a few do not pursue learning by design.
- The types of learning implemented in the cognitive architectures span a variety of approaches that largely depend on the type of knowledge acquired: declarative learning is mostly done through symbol manipulation, whereas perceptual and procedural learning use multiple techniques, including supervised, unsupervised, and reinforcement learning, in addition to symbolic methods.
- Procedural reward-based learning is the most common since it requires minimal to no supervision and mimics processes found in living organisms.
- Although a sizable group of cognitive architectures do not implement any learning, there is no cognitive architecture that is incapable of any learning in principle.

7 Reasoning and Decision-Making

Reasoning and decision-making are often associated with logical inference, problem-solving, weighing possibilities, and other intellectual activities. In reality, they extend to practically any information-processing that helps decide "what to do next?" When the answer is obvious, the decision is made automatically with little effort. More often than not, some deliberation may be needed to choose among many alternatives, without full information, under time pressure, and in uncertain conditions. In this chapter, we will discuss these and other aspects of reasoning and decision-making and how they are implemented in cognitive architectures.

Section 7.1 details current views on defining cognition, thinking, reasoning, and decision-making.
Section 7.2 discusses the types of theoretical reasoning (or reasoning about beliefs), focusing on everyday non-monotonic reasoning as a mechanism for dealing with incomplete and uncertain information.
Section 7.3 is dedicated to practical reasoning or decision-making. Here, we start with the aspects of physical action selection that are most important for producing appropriate and consistent behaviors. Then, we discuss how biological and artificial mechanisms for action selection implement these features. In the final subsection, we consider various behavior modulators, such as personality traits, drives, and emotions, which affect decision-making.
Section 7.4 covers the most common meta-reasoning mechanisms in cognitive architectures, such as self-monitoring and self-regulation.

7.1 What is reasoning?

Cognition, thinking, reasoning, and decision-making are familiar terms that are often used in the same context as descriptions of human mental processes, but what exactly these processes entail is often left unsaid.

Cognition. There is a spectrum of definitions of cognition. According to Bayne et al. (2019), a narrow view limits cognition to operations on language-like representations (beliefs, desires, and intentions). A slightly broader understanding is that any information processing that is not associative can be called cognition. However, associations themselves can be divided into sub-cognitive (habituation and classical conditioning) and cognitive (attention, prediction, and intentionality). Because the former give rise to the latter and both share the same neural pathways, the boundary between them is not clear. Another proposal draws the line between perception and cognition. It has recently become a point of discussion again courtesy of Firestone and Scholl (2016) who analyzed a large volume of experimental data to disprove the notion of "top-down" effects that cognition exerts on

even finer distinction and exclude only low-level perception. For example, the term "visual cognition" has been applied to mid- and high-level vision that performs inference on retinal representations which are the result of low-level vision (Pinker, 1984; Cavanagh, 2011).

Finally, some suggest that defining cognition is neither necessary nor useful and propose to instead treat it as an umbrella term for any information-processing that can be expressed in computational terms (Allen, 2017). Bayne et al. (2019) go further and suggest abandoning the term "cognition" altogether in favor of a new vocabulary not saddled with the outdated associations.

Thinking. The scope of thinking also has several interpretations depending on the discipline. The philosophical investigations of thinking historically focused on rational discourse, argumentation, and logical inference (Lewis and Smith, 1993). But other opinions also exist. For example, Paivio (1975) considered a dual system of thinking consisting of parallel and heuristic perceptual processing and verbal information processing, which is sequential and analytical. Some psychologists extended the notion of thinking to nearly all known psychological phenomena in humans (Oden, 1987) and animals (Griffin, 1984). In this broadest sense, thinking is synonymous with cognition as in any mental activity, including perception, categorization, decision-making, etc. (Rips, 1990).

A narrow view of thinking restricts it only to the so-called high-level cognitive operations, such as decision-making, judgment, reasoning, and other verbal activities (Holyoak and Spellman, 1993). Thinking skills may be subdivided further into lower and higher order, depending on how the knowledge is applied. Lewis and Smith (1993) describe this as the difference between reproductive and productive behavior, respectively. Reproductive behavior is an extraction of patterns from previous experience and their direct application in the same circumstances (e.g. learning a multiplication table and applying it to multiply two numbers). Productive behavior (or reasoning) is equated with problem-solving with no or little previous exposure to similar problems and in a new challenging context. However, even if considered as a high-level ability, thinking is still connected to and affected by perception and motor functions (Watson, 1920).

Reasoning and decision-making. Various definitions of reasoning likewise attribute to it a wide range of abilities. Some consider reasoning equivalent to thinking, e.g. reasoning as infiltrating all thought (Rips, 1990), involved in any type of symbolic manipulation (Levesque, 1986), or as any mental process that leads from one mental state to a new one (Fink, 2013). Others restrict reasoning only to conscious and deliberate revision of beliefs and attitudes following specific rules (not limited to rules of logical inference), thus excluding automated, intuitive, or one-step mental processes (McHugh and Way, 2018). A more principled definition by Galotti (1989) describes four characteristics of reasoning as 1) a mental activity that 2) transforms premises to reach conclusions or goals 3) according to rules of logic when possible, which 4) may lead to the modification, replacement, or deletion of the initial premises.

The distinction between reasoning and decision-making is often framed as the difference between theoretical and practical reasoning. An intuitive definition for these terms commonly found in the literature describes theoretical reasoning as concerned with beliefs and practical reasoning with everything

else, including actions and intentions (Pollock, 1989). In other words, theoretical reasoning expands a view of the world by incorporating new evidence into existing knowledge in a coherent way while following certain rules of inference. These rules need not be logical, but must be non-trivial, e.g. accepting or rejecting all observations or doing so at random is not considered reasoning (McHugh and Way, 2018). Practical reasoning likewise seeks to adopt intentions that are consistent with known facts and, ideally, satisfy one's desires (Harman, 1976). The two processes are intertwined: practical reasoning may require theoretical reasoning to explain past actions, and theoretical reasoning may lead to forming beliefs that lead to new intentions. Omitting for now the specifics of the mechanisms involved, the key points that are universally agreed upon are that both types of reasoning are similar but are applied to different premises and lead to different outcomes (Johnson-Laird and Shafir, 1993; Legrenzi et al., 1993; Alvarez, 2010).

The following relationship between the terms discussed above can be established based on this short review: cognition $\supseteq$ thinking $\supseteq$ reasoning $\supseteq$ decision-making. In most cognitive architectures, reasoning is treated as a deliberate "high-level" cognitive ability. Few depart from this notion. For example, Copycat, DUAL, and LISA extend reasoning to all aspects of cognition, from low-level (perception) to high-level (planning and language understanding) (Hofstadter, 1984; Kokinov, 1988; Taylor and Hummel, 2007). SHRUTI focuses on reflexive reasoning, an unconscious and spontaneous inference-making, also at odds with one of the definitions given earlier (Shastri and Ajjanagadde, 1993).

With respect to the theoretical/practical reasoning divide, the majority of cognitive architectures pursue either one or the other, with a select few capable of demonstrating both. As will be discussed in the next sections, the inference rules and mechanisms are generally the same, supporting theoretical conclusions in practice.

7.2 Reasoning about beliefs

Our beliefs about ourselves and the world around are constantly evolving, shaped by new experiences, observations, and interactions. Yet, we are able to form a relatively coherent picture out of these disparate sources of information. In this section, we will discuss what role logical inference plays in this process and how it is modeled in cognitive architectures.

7.2.1 Monotonic and non-monotonic reasoning

Colloquially, reasoning is strongly associated with logic and rationality. Although both play a role in reasoning, not all reasoning is logical or rational, therefore logic alone cannot provide a complete theory of reasoning (Harman, 1976).

Formal logical reasoning presumes the completeness of the premises and their truth, however, both conditions are violated in daily life. Often, everyday beliefs do not reflect a full understanding of the situation nor all possible effects of possible actions. This inherently uncertain and hypothetical nature of beliefs calls for their continuous re-evaluation every time new evidence

becomes available. As a result, existing beliefs can be extended, modified, or overturned, and new beliefs can be formed. This is referred to as *defeasible* or *non-monotonic* reasoning.

The standard example of non-monotonic reasoning in classical AI is this: given the statements that "Birds can fly" and "Tweety is a bird," one is justified in concluding that "Tweety can fly." However, this conclusion may be invalidated by additional statements about Tweety, e.g. "Tweety's wing is broken" or "Tweety is a hatchling."

In contrast, during *monotonic* reasoning, neither the premises nor the conclusions derived from them using formal rules can be invalidated by adding new premises. For example, a statement "$2 + 2 = 4$" does not change upon addition of other statements like "Sun sets in the West" to the knowledge base. A typical example of monotonic reasoning is deduction.

Cognitive architectures are naturally concerned with non-monotonic reasoning in the domains with limited information. Many common tasks, such as planning, making arguments, and choosing among many options do not operate with full information about the state or effects of actions. These have to be hypothesized or gradually filled in through experience. The information limitations apply to cognitive architectures regardless of whether they model human everyday reasoning or optimal rational behavior. A number of techniques fall under non-monotonic reasoning: probabilistic reasoning, analogical reasoning, case-based reasoning, explanation-based reasoning, decision-theoretic reasoning, and inductive reasoning (Pollock, 1987; Chen, 1991; Loui, 1993).

7.2.2 Types of logical inference

Logical inference is a basic and well-studied form of reasoning. The most fundamental types of inference are defined following Peirce's taxonomy (Staat, 1993):

Deduction is an application of rules of inference to the given premises to reach a conclusion. In a valid deductive argument, the conclusion necessarily follows from the premises. As such, it is the only type of reasoning that does not violate monotonicity.

Induction, unlike deduction, is approximate; the conclusion is supported by the given premises but does not necessarily follow from them.

Abduction is usually described as finding the best explanation of the given premises. Like induction, it is approximate and is not entailed by the premises.

Analogy is a comparison between two entities that seeks what makes them similar.[1]

Valid methods of inference and rational thinking have been studied for centuries in logic and philosophy, and a variety of formal representations and rules were discovered for performing these operations and explaining their properties. Research on psychological and biological aspects of human reasoning is relatively recent in comparison. Over the past decades, mounting

[1]Even though analogy was originally excluded from Peirce's taxonomy because it was thought to be a compound of induction and abduction, we list it here due to its importance in human thought and attention it received in cognitive architectures.

Table 7.1 Examples of the cognitive architectures exploring different aspects of theoretical reasoning.

Cognitive architecture	Focus
Copycat/Metacat	analogy-making
Companion	analogy-making
LISA	analogy-making
SHRUTI	reflexive reasoning
OSCAR	defeasible reasoning
Disciple	explanation-based reasoning
DUAL	deduction, analogy-making
NARS	reasoning with insufficient knowledge and resources

psychological evidence shows that people often behave and think irrationally in practice (Cohen, 1981; Shafir and LeBoeuf, 2002). However, despite multiple documented instances of human subjects failing to reason logically, the prevailing opinion is that humans are not inherently irrational but rather fail to apply proper inference procedures in practice (Stein, 1997; Stanovich, 2012).

There is less agreement on what mechanisms better explain observed faults in human reasoning behaviors. Two main theories of how people reason are formal rules and mental models (Roberts, 1993):

Formal rule theory suggests that humans possess a natural logic system which first transforms the information into abstract logical premises and then applies appropriate rules to reach conclusions. The inconsistencies in human reasoning can thus be explained by the number and availability of such rules (Rips, 1983; Braine and O'Brien, 1991).

The theory of mental models proposed by Johnson-Laird and Byrne (1993) does not assume that there is a natural logic system in place. Instead of manipulating symbolic propositions according to some rules, humans form mental models (internal representation) of the problem situation in working memory (WM) and modify it as new evidence comes in. Human reasoning proceeds in three stages:

1. Build a mental model (or extend an existing one) with the given premises, taking into account general knowledge;
2. Draw a provisional conclusion that is not a trivial inference nor a repetition of the given argument;
3. Evaluate the conclusion by considering counterexamples, and if none are found, assume the conclusion to be valid.

Unlike the formal rule theory, the mental model theory does not draw an explicit distinction between deduction, induction, and abduction since all of these types of reasoning are intertwined in the process of continuous updating of the internal model and drawing conclusions (Johnson-Laird, 2010).

This distinction is generally not upheld in cognitive architectures: those concerned with theoretical reasoning (listed in Table 7.1) model some aspects of human performance, but only in Companion are they explicitly associated with the theory of mental models (Friedman et al., 2009). Similarly, deduction, induction, and abduction are not modeled separately, although the mechanisms for these types of inference can be found in virtually all architectures. Rather, each architecture focuses on a different aspect of reasoning. For example, DUAL emphasizes human-like inference and SHRUTI demonstrates the viability of the connectionist approach to deductive and inductive reasoning.

Several cognitive architectures focus specifically on non-monotonic reasoning. For instance, OSCAR is a general reasoning architecture implementing an epistemological view of defeasibility. Disciple also examines evidential reasoning and mechanisms for determining the relevance and credibility of premises and conclusions. NARS is another reasoning engine that like the previous two focuses on establishing the validity of beliefs stemming from lack of complete knowledge but also takes into account limited resources for reasoning.

Of all reasoning types, analogical inference received the most attention and will be discussed separately in the next section.

7.2.3 Analogical reasoning

Analogical reasoning is broadly defined as drawing similarities, noticing resemblances, and establishing correspondences across different sets of relations and properties. The ability to compare and notice similarities is considered one of the fundamental cognitive mechanisms that, besides analogical reasoning itself, is involved in perception and memory retrieval Hofstadter (2001), is a prerequisite for other kinds of logical reasoning (Sowa and Majumdar, 2003), and is integral for the acquisition of representational systems, such as language or mathematics, and associated skills (Vosniadou, 1995).

In everyday life, analogy is crucial for learning and adaptation as a way of transferring knowledge from the past experience to new situations. For example, a person who learned to drive a specific car can fairly easily switch to a different one. Analogy is also useful for explaining concepts that cannot be experienced directly, e.g. "electricity flows like water" (Gentner, 2003).

Overall, there are many types of analogies and many ways of categorizing them: by purpose (Waller, 2001), as within-domain or across-domain (Burstein, 1989), or by the degree of similarity (Gentner, 1983). However, all kinds of analogies require the same basic steps that are in agreement with the psychological literature and found in most computational models (Hall, 1989; Kokinov and French, 2003; Gentner and Forbus, 2011):

- *Encoding*—building a suitable representation for the given input (target);
- *Retrieval (recognition)*—retrieving potential similarities, referred to as source or base, from long-term memory;
- *Mapping (transfer)*—establishing correspondences between the elements of the source and target;
- *Evaluation*—checking soundness of mapping to justify, repair, or extend it;
- *Consolidation (abstraction)*—saving the results for the future use.

The steps above describe only the high-level phases gleaned from human psychological data, and can be further subdivided into multiple operations and processes. To date, dozens of theoretical and computational models have been proposed that omit, extend, combine, or interleave these basic steps (for an overview, see Hall, 1989 and French, 2002). Below, we discuss several influential analogy-making models that have been integrated into cognitive architectures and the cognitive architectures that investigated analogy-making (Table 7.2).

Structure Mapping Engine (SME). SME implements Structure Mapping Theory (SMT) Gentner (1983), which remains one of the most influential theories of analogy-making to date. As its name suggests, the theory emphasizes the structural correspondence between the source and the target

Table 7.2 Symbolic, connectionist, and hybrid models of analogy-making and the cognitive architectures that incorporate them. LISA and Copycat/Metacat include a model of analogy as part of their architecture.

Representation	Analogy model	Cognitive architecture
Symbolic	SME derivational analogy	Companion PRODIGY
Connectionist	LISA	
Hybrid	AMBR Copycat/Metacat	DUAL

domains. According to SMT, the domains and situations can be thought of as systems of objects, their attributes, and relations to other objects. Objects, attributes, and relations are represented as a propositional network of nodes and predicates. An analogy "T is like a B" is defined as a mapping between domain B (base) and T (target) that follows simple rules: a) the attributes of objects are discarded; 2) the relations between the objects are preserved. The key here is that these rules are structural, i.e. they rely on syntax rather than the content of the statements. Based on these principles, SME computes a set of global interpretations of the comparison between the base and the target which contains: correspondences (matches between inputs), structural evaluation (soundness of the match), and candidate inferences (what follows from the match) (Forbus et al., 1995).

Since SME on its own can only perform the mapping, another component was added to incorporate the model into the Companion architecture. This component, MAC/FAC ("many are called, but few are chosen"), performs a similarity-based retrieval that provides candidates for mapping (Forbus et al., 1995). In the first stage, called MAC, the long-term memory is probed by the cheap matchers that quickly return many candidates that are superficially similar to the base case stored in the WM. The second stage, FAC, runs in parallel a number of SME matches which compute the structural alignment between the base and the candidates and threshold the results to select the most promising ones.

Derivational analogy. Unlike SMT, the derivational analogy approach proposed by Carbonell (1983) is not psychologically motivated and focuses on the integration of analogy-making to improve problem-solving. Analogy is operationalized accordingly: two problems are considered similar if they share the initial reasoning steps; given a problem, reasoning steps from the past solutions can be transferred and adapted to a new situation; the adaptation process retains the past steps that apply directly to the new situation and replace or modify those that do not. To enable this behavior, past solutions must be retained in the memory, complete with intermediate steps, and reasons for their success and failure.

Derivational analogy can be seen as a generalized form of case-based reasoning and complementary to explanation-based learning (EBL) (Veloso and Carbonell, 1993). Case-based reasoning applies analogy in a limited way for finding the most relevant case out of the library of past cases and adjusts it for application in the new scenario. EBL is a form of generalization that acquires general knowledge from several specific examples (DeJong, 1988), which can be very helpful for retrieving analogies. PRODIGY takes advantage of both by combining EBL with the derivational analogy (Veloso, 1994) to relax the

requirement of complete domain knowledge and enable flexible reuse of past learning episodes.

LISA is a model of relational reasoning which uses a localist neural network to represent the relations between entities in a distributed fashion: objects, attributes, and types of relations are nodes in the hierarchical network connected by the edges with activations of varying strength (Hummel and Holyoak, 2005). Unlike the previously discussed models, LISA interleaves retrieval and mapping: a new problem (target) in working memory (WM) triggers cueing of the base case from the long-term memory by firing the propositions that generate patterns of activations. These in turn retrieve analogs back to the WM for mapping. The units that fire together are reinforced using a simple form of Hebbian learning (Hummel and Holyoak, 2003).

In LISA, retrieval, mapping, inference, and schema induction are performed with the same mapping algorithm and self-supervised learning algorithm. In addition, the capacity of its WM is limited, therefore only three mappings can be made at a time (Hummel and Holyoak, 2005). In comparison, other models place no limitations on the size of the WM and number of candidates that can be retrieved.

Copycat is an architecture that explores creativity in analogical reasoning. Analogy-making is represented internally as the interaction of multiple concurrent agents (codelets) that connect perception, top-down influences from long-term memory (a semantic network called Slipnet), and bottom-up pressure from WM (referred to as Workspace). A semantic network with concepts stored in the nodes and activations propagated along the edges helps capture semantic context (halo) of the concepts and implement "slippage" from one related concept to another, useful for retrieval. Much of the behavior of Copycat is emergent; the solution "bubbles up" through iterative strengthening and weakening of the links between the concepts.

The model also includes built-in preferences that nudge its behavior in the desired direction. One is an inclination to seek "deeper" analogies rather than superficial similarities. Another is stochasticity, expressed as a temperature, which is kept high in the beginning to encourage the diversity of candidate solutions and "cools off" when patterns begin to emerge within the set to encourage convergence on the best solution (Hofstadter, 1994). Copycat does not model the consolidation stage, as it is reset between every run.

What distinguishes Copycat from other models is that it builds its own representations from raw human-readable input and combines the process of encoding with retrieval (Hofstadter and Mitchell, 1994). However, it also operates in the most restricted micro-domain of 26 letters, so the raw inputs are fairly easy to parse. All inputs are formatted in the same way, e.g. "$abc \rightarrow abb; xyz \rightarrow ?$," where the first expression is an example of a transformation from abc to abb and the second expression asks how xyz can be transformed in the same way. The output is the resulting pattern, e.g. xyy.

AMBR is a model of analogical reasoning based on the DUAL cognitive architecture (Kokinov, 1994b). As it is the most recent of the models discussed here, it combines many of the features of other models. It has similarities to Copycat in that analogical common-sense reasoning emerges from the interactions between multiple agents, although the structure of the agents is different. In AMBR, each agent has a symbolic part that encodes declarative/procedural

knowledge and a connectionist part that computes the activation level associated with relevance of knowledge. Similarly to LISA, retrieval and mapping are a result of the interactions of multiple agents, but with no limitations as to how many can operate at a time.

7.3 Reasoning about actions

The process by which complex behaviors arise is studied across many disciplines. In psychology and neuroscience, it is framed as behavioral choice, motor program selection, or decision-making (Prescott, 2008), whereas in AI it has been referred to as the action selection problem (Maes, 1989). In general terms, the process of choosing what to do next can be described as an infinite three-step loop (Hayes-Roth, 1985a): 1) identify a set of next possible actions; 2) select the next action from the set; 3) execute the action.

The assumptions often made in the literature are that behavior is a) directed by a goal or is a response to an event and b) can be recursively subdivided into chunks, the smallest of which are called actions. Not all actions should result in a motor movement; some may not even have any external expression, but instead set a new goal to pursue (Pirjanian, 1999).

Both goals and actions can be considered at different levels of abstraction. On the highest level, most biological organisms pursue propagation of their genes and survival (Dawkins, 1976). Ethology recognizes the groups of functionally related behaviors that help reach these goals, such as body maintenance, feeding, reproduction, care for offspring, etc., which themselves can be further discretized (Tyrrell, 1994). Neurobiological analysis of behavior also functionally decomposes it into multiple levels of selection, where higher levels decide on the goals and patterns of activities to reach them, and the bottom level converts these decisions into muscular activations (Redgrave et al., 1999). Similarly, in AI, action selection encompasses mental activity (setting goals and forming beliefs) and physical acts (moving a joint) (Öztürk, 2009).

In sum, the consensus is that in both biological and artificial systems action selection is modular and hierarchical (Öztürk, 2005) with the basic selection loop replicated across modules and levels of decision-making. What constitutes atomic action, however, is context-dependent. Even in everyday speech, we can describe the same action in different ways. For example, grabbing a cup can be considered a single act or decomposed into three physical actions of reaching, grasping, and moving. On a higher level, one can also include a feeling of thirst and forming an intention to drink preceding the physical action. On the lowest level, these actions are further decomposed into motions of the individual joints and neurons that drive them. In most embodied artificial systems, atomic elements usually refer to discrete physical actions at the "middle" level, but, ultimately, the number of primitives and their resolution depends on the design of the architecture and the task.

7.3.1 Selecting the best action

There is a general agreement across multiple disciplines as to what constitutes a good choice of action. The following list of desirable features for action

selection is compiled from multiple sources (Simon, 1956; Tyrrell, 1994; Pirjanian, 1999; Redgrave et al., 1999; Brom and Bryson, 2006; Prescott, 2008):

Appropriate. The most obvious criterion is that the action should be appropriate for the external circumstances and intrinsic demands. The probability of selecting a certain action should increase if it is more relevant to the current situation and has a better chance to satisfy a need. If there are multiple factors that motivate the use of the action, it should receive even higher support.

Sufficient. The selected action does not necessarily have to be optimal, merely good enough. This is also referred to as satisficing.

One action at a time. This applies to both mental and physical actions. While a system can maintain multiple goals, it must select one to pursue at any given moment in time. Similarly, a single action must be chosen among the alternatives competing for the same effector.

Timely. Decision on taking an action and the physical action itself should be commensurate with the rate of change in the environment to increase the robustness of the system to dynamic and unpredictable changes.

Persistent. An action, once taken, should be maintained for a sufficient amount of time. Otherwise, if there are alternative actions with similar relevance, the system may start to oscillate between them, diminishing the effect of the action.

No interference. Other actions should not interfere with the selected action by canceling its effects or invalidating the conditions for its operation.

Interruptible. An action selection mechanism should allow interruptions to respond to urgent contingencies or switch to the action with better support should it become available.

Clean switching. Selecting an option among alternatives should be quick and decisive to avoid idleness after the previous action was completed or interrupted.

It is not difficult to notice that some of these requirements are contradictory: persistence is at odds with interruptibility since timeliness restricts the amount of time available for assessing the appropriateness of actions. For example, an action that has immediate benefits may be worse in the long term and a suboptimal action should be taken instead. One can also imagine a situation where deliberation or the action cannot be completed in time and must be halted. Therefore, a mechanism is needed for determining which situations demand an urgent response and which leave time for more deliberation. In the next section, we will discuss how the existing action selection mechanisms propose to address these issues.

7.3.2 Dimensions of difference

There are several dimensions of differences that are often used to delineate approaches to action selection in biological (Öztürk, 2005) and artificial systems (Pirjanian, 1999). Here, we will consider the three most common ones: reactivity vs. planning, hierarchical vs. non-hierarchical, and competitive vs. cooperative coordination.

Reactivity/deliberation

In artificial intelligence (AI), action selection mechanisms can be seen as spanning the spectrum between fully reactive and fully deliberative approaches (Rosenschein and Kaelbling, 1989; Wood, 1995; Bryson, 1999). The latter is a classical AI paradigm that views behavior generation as a two-step process—planning followed by execution.

Typically, planning requires three inputs: 1) the initial state of the world and the agent, 2) the agent's goal, and 3) a set of actions and constraints (domain theory). Given that the inputs are correct, a planner then outputs a sequence of actions that will take the agent from the initial to goal state. The execution of the plan is trivial—the actions are interpreted and applied to the world to achieve the desired result. The key theoretical issues with planning are its computational intractability for most non-trivial tasks (Chapman, 1987) and unfeasibility of constructing a complete domain theory (Ginsberg, 1989). The key practical issue with planning is that it takes time, therefore unexpected events that outpace planning or occur during planning cannot be addressed. Even a minor change in the external conditions will prompt a new cycle of deliberation, potentially stalling the system indefinitely. This limits the application of planning to the static or highly predictable environments where the agent is the primary driver of changes.

On the opposite end of the spectrum is reactive control, which foregoes explicit reasoning and prescribes behaviors directly by mapping sensory input onto motor commands. Because no deliberation and no state maintenance is necessary, an action can be chosen much faster. This is an advantage in quickly changing environments, however, lack of deliberation also makes anticipating future events impossible.

While a complex system constructed entirely out of reactive actions is theoretically possible, it is rarely done in practice. The Subsumption architecture that pioneered this approach showed that a purely reactive system could give rise to interesting behaviors, mostly limited to navigation and following (Brooks, 1990). However, it was quickly discovered that most non-trivial tasks cannot be done without deliberation and memory (Hartley and Pipitone, 1991).

Therefore, the vast majority of cognitive architectures use a mixture of reactivity and deliberation in deciding their next action. Reactive actions are represented as the deterministic sensorimotor processes that are executed without deliberation. In this sense, they are somewhat analogous to reflexes in biological systems. There are many examples of reactive behaviors in cognitive architectures, most of which fall into the following categories:

- *Protective.* Prevent damage to the body by using special devices (e.g. touch sensors) or limiting the range of motion. These are naturally found in embodied systems.
- *Basic motor actions.* Basic primitives for moving body parts. These primitives are platform-dependent as well.
- *Innate responses.* Predetermined preferences are not always designed to protect the body, and can also serve as a scaffolding for learning more complex skills or arbitrary personality preferences.
- *Learned shortcuts.* As the system interacts more with the world, some actions may be learned to be invoked automatically.

Off-line planning is also used in several cognitive architectures, but as a high-level guidance rather than as a means of specifying all steps. This approach received the somewhat contradictory name of reactive planning and was widely popular in the 1990s across a number of robotic architectures, such as 3T, ATLANTIS, CIRCA, PRS, and TCA. All of these architectures use classical state-based planners to provide high-level guidance for Subsumption-like reactive components that convert high-level behaviors into motor actions.

Nearly all remaining cognitive architectures implement dynamic action selection, where high-level reasoning and the decomposition of tasks into smaller chunks are tightly interleaved with low-level dispatch of motor actions. Blackboard and production architectures and their variations are obvious examples. In these systems, at every decision cycle, available actions are matched against the current context and one (or sometimes more) is selected for execution (the criteria for selection are discussed below). The action may be a motor command or a high-level operation, such as updating a goal or creating a new one. Thus, interleaving the task decomposition and execution of primitives ensures that the decisions are up to date with the external conditions and the internal state.

Hierarchical/non-hierarchical

Another dimension along which decision-making mechanisms are differentiated is their representational organization, with hierarchy being an important type. According to a classical definition by Dawkins (1976), two entities A and B are hierarchically related if A "is a boss" of B. "Boss-ness" broadly means that A is superior to B (e.g. A has a direct or indirect causal effect on the state of B). The definition is recursive; any boss of A is also a boss of B and any element of which B is a boss is subordinate to A as well. Thus, hierarchy is a set, which satisfies two conditions: 1) no element in the set is superior to itself and 2) one element is superior to all others.

The simplest hierarchy is a linear one (e.g. $A \rightarrow B \rightarrow C$) but most tree-like structures are also considered hierarchical. Cyclic structures and networks are typical examples of non-hierarchies, but for different reasons. Cyclic structures are non-hierarchies because there is no top node, while most networks are non-hierarchies because of multiple top nodes. One can argue, of course, that at a different abstraction level networks can be considered overlapping sets of hierarchies or that layers of neurons, not the individual units, form a hierarchy. Finally, the presence of feedback connections, particularly those linking distant elements, disrupt the hierarchical relationships.

Various types of hierarchical structures have been found in human behavior and their underlying neural representations (Dawkins, 1976; Botvinick, 2008; Raut et al., 2020). Hierarchical organization is also common in artificial systems (Wilson, 1979; Merel et al., 2019). Cognitive architectures are no exception, however, deciding whether any given architecture has a hierarchical or flat control structure is not easy. There are several reasons for this. First, decision-making mechanisms are often not labeled as hierarchical or otherwise by the authors. Anyone attempting to deduce such labels from the written descriptions will often find them difficult to locate and incomplete, especially if decision-making is not the focal point of the architecture. Furthermore, depending on the level of abstraction, the same mechanisms may be interpreted as hierarchical or flat (e.g. neural networks mentioned earlier). The same

concern applies to graphical depictions of cognitive architectures, which also may impose a hierarchical structure where there is none or fail to show it when it exists. Finally, the hierarchical structure of the architecture itself or the hierarchical representation of the procedural knowledge can be conflated with hierarchical control.

Let us consider several examples to illustrate the difficulties in characterizing the hierarchical nature of control. For instance, in the Subsumption architecture, action selection can be viewed as hierarchical (Epstein, 1992b). The reason for this categorization is that individual behaviors are organized such that more complex behaviors at the top "subsume" (override) the simpler ones below. And yet Subsumption is more often seen as non-hierarchical (Brooks, 1985; Brooks, 1991; Byrnes et al., 1992; Spector and Hendler, 1994; Öztürk, 2005) because higher level behaviors do not directly call lower ones, but rather suppress their independently made selections.[2]

Another representative example is the three-layer architecture. Here, the hierarchical structure is also noticeable; the deliberative layer plans the next steps that guide the scheduling layer in the middle and the bottom reactive layer, which invokes the motor commands. The 3T and ATLANTIS both adopt this approach (Gat, 1998) and are described as having a control hierarchy (Schreckenghost et al., 1998; Pirjanian, 1999). However, the authors of ATLANTIS claim that its decision-making is not hierarchical (Gat, 1991b). Instead, all decision-making happens in the middle layer, which monitors progress, commands the reactive layer to act, or prompts the deliberative layer for more planning. The RCS architecture combines the elements of Subsumption and layered architectures within the pronounced hierarchical structure (Albus, 1991). The modules of the system are organized in layers with defined functional roles (from high-level planning to reactivity) and resolution (spatial and temporal). Each module itself is hierarchical as well, consisting of a high-level planner, the plan selector, and executor. The functioning of the system, however, is closer to Subsumption in the sense that each module operates independently, and the overall behavior is coordinated via nested feedback loops. FORR demonstrates yet another variation on hierarchical structure (Epstein et al., 2010). Here, individual modules called advisors are organized in three tiers. The tier-1 advisors are reactive and guaranteed to be correct. If they cannot respond to a situation, the advisors from tier 2 above are consulted and produce a set of options. If a selection cannot be made, then the tier-3 advisors vote to make the final choice.

On the other end of the spectrum is the non-hierarchical control structure. Perhaps, the best known example is Maes's (1991) Agent Network Architecture (ANA), which is organized as a recurrent non-hierarchical network with nodes representing reactive actions. Each node has a set of preconditions that define when it can be activated and links to other nodes. At every cycle, the sensory information and the internal state are fed into the network. Those nodes whose preconditions match the input are activated and spread the activation further

[2]Tyrrell (1993) even argues that subsumption is not an action selection mechanism but rather a computational substrate that can implement any selection algorithm. In truth, the same can be said about most cognitive architectures. However, in case of the Subsumption architecture, its many instances built for various robots tend to have a similar structure. Likewise, most cognitive architectures tend to commit to certain representations and mechanisms even though they can accommodate other options.

along the successor and predecessor links. An inhibition signal is spread along the conflictor links. Finally, the one with the highest activation is executed. Then, the cycle repeats. If no node can be executed, matching criteria are gradually relaxed. This organization is highly modular and robust, which is an advantage in many applications. Copycat is another example of largely non-hierarchical decision-making. In Copycat, actions and decisions emerge from the interaction between multiple computational agents (codelets). Which codelet's output is selected is decided dynamically, thus none are prioritized over others by default (Hofstadter, 1984).

The mechanisms introduced in ANA and Copycat are found in many architectures but almost always mixed in with some hierarchical elements. For example, VMattie combines ANA and Copycat, and Kismet, LIDA, ERE, Ymir, and IMA mix the behavior networks with the Subsumption-like hierarchy. The production systems, such as ACT-R, Soar, EPIC, and CAPS, also have elements of non-hierarchical control. The production rules in these systems are analogous to the nodes in ANA or the codelets in Copycat, and likewise respond to input and output actions if selected. Activation spreading is generally done through the memory and in some cases is mediated by a goal stack.

In sum, hierarchical and modular structures are found across all cognitive architectures but very few implement strictly hierarchical or strictly flat control. Even when decision-making is organized according to levels of task abstraction, spatio-temporal resolution, and reactivity, feedback loops are introduced that break the hierarchy. Likewise, flat non-hierarchical control structures often have an implicitly hierarchical organization. For example, links between the modules can specify successor-predecessor relationships (which may be temporary and dynamically adjusted). In addition, hierarchical knowledge representation can be used to guide decision-making from goal-setting to motor command execution.

Competitive/cooperative coordination

Different subsystems connected to the same actuators may request control at the same time. The outcome and the resulting action depends on what type of conflict resolution mechanism is employed. Strictly competing mechanisms, such as priority-based or winner-take-all, will allow only one action to take control or a single goal to be prioritized. This scheme does not allow multiple non-conflicting actions or goals to be achieved. Cooperative coordination of behaviors combines contributions from multiple actions and reaches an action or actions that represent consensus between them.

Overall, arbitration is more common than coordination and the difference between approaches is mainly based on the criteria for selection as listed in the next section. An example of cooperative action selection is Copycat, where many concurrently running agents contribute to the final solution with a global mechanism controlling the amount of stochasticity in their behaviors (Hofstadter, 1984). Along similar lines, the blackboard-based architectures combine contributions from multiple knowledge sources, which respond to different events and influence one another through the shared information (Hayes-Roth, 1985a).

7.3.3 Action selection criteria

Regardless of the action selection mechanism, decisions are always made based on a set of criteria. Below, we will discuss some of the most common ones.

Relevance is the most basic condition for action selection, regardless of the abstraction level. Suitability of the action can be determined as relevance to a) the goal/task, b) the external situation, or c) some combination of the two. However, relevance alone is often not sufficient for choosing a single action to execute, as two scenarios may potentially occur:

- *Many actions are relevant.* In this case, additional criteria are needed to increase chances of a match or modification to the goal.
- *No actions are relevant.* In this case, the system may relax the matching conditions, backtrack, or abandon the goal altogether.

Utility of the action is usually determined by its past use. For example, in ACT-R, if more than one rule matches the contents of the working memory (WM), the one with the higher probability of success is selected. To avoid selecting the same actions, there is a small chance that a less optimal action will be selected to encourage exploration.

Urgency is defined as a temporal limit on the execution of an action (e.g. running out of battery or responding to a concurrent event), however most cognitive architectures do not consider the influence of time on task performance.

Priority is a more general concept, which can be computed using some or all criteria listed above. Another common solution is to predefine priorities for certain actions or high-level goals. This by design happens in the hierarchical systems where higher level layers of control have higher priority than the lower ones (e.g. Subsumption, RCS). In some architectures, the priorities of high-level goals are hard-coded (e.g. ARCADIA, CERA-CRANIUM).

7.3.4 Behavior modulation

That behavior modulators, such as emotions, moods, feelings, and personality traits, play a role in decision-making has been known for most of the 20th century. It is now believed that emotions may even supersede rational deliberation in their influence. However, this line of research only saw significant uptake beginning in the 2000s (Lerner et al., 2015).

Modeling emotion in cognitive architectures followed a similar timeline, thus most of the earlier architectures lacked any mechanisms for it. While any cognitive architecture can be "retrofitted" with emotions, it is rarely done. Overall, only one-third of the projects we considered model internal motivation, many of them relatively recent. A notable exception is Soar, which added support for emotional decision-making in version 9 (Laird, 2008). Although many affective augmentations were proposed for ACT-R, they have not yet been made a part of the core architecture. Across cognitive architectures, three major components of behavioral modulation can be found—drives, emotions, and traits—which will be discussed below in more detail.

Drives

The concept of drive has a long history, originating in the early studies of instinctual animal behavior (Lorenz, 1935; Tinbergen, 1951). In humans, drives

are also considered as innate factors that motivate behavior and bias it in a certain direction (Brigandt, 2005). Usually, drives represent basic survival and reproduction needs, but there are also secondary drives that may be induced by external stimuli, such as social, thermal, and chemical (Burghardt, 2019). The sense in which the term "drive" is used in the AI literature is as an innate basic physiological and social need, e.g. body maintenance, social interaction, self-preservation, etc. Many drives, such as rest and feeding, have a cyclical and homeostatic nature, meaning that they slowly increase over time and exert more and more bias toward the actions that bring them back to the original value.

Drives are generally not common in cognitive architectures. Kismet, MicroPsi, and Clarion are the only architectures where drives play a significant role. In particular, Kismet's entire behavior is controlled by three drives: social, stimulation, and fatigue (Breazeal, 2003a). Each drive is associated with accompanying consummatory behavior(s), such as playing with people, playing with toys, or sleeping. Without appropriate stimulation, drives gradually rise until satiation. Drives perform three functions: 1) bias behavior selection toward actions that satiate the drive, 2) generate a goal which organizes behavior and perception, and 3) influence the robot's affective state (further discussed in the subsection below).

MicroPsi refers to drives as demands of the system associated with specific urge signals. These urges and corresponding drives are grouped into three categories: physiological (fuel, water, intactness), social (affiliation), and cognitive (certainty, competence) (Bach, 2011), although a broader range has also been proposed (Bach, 2012b). Similar to Kismet, urges are fixed and hardwired as parameters that tend to deviate from their target values and are maintained within an acceptable range by promoting certain behaviors. What is different here is that the behaviors and situations that satisfy needs are not predefined, but are learned instead via reinforcement learning (Bach, 2011).

Another example is Clarion, which separates basic physiological needs (food, water, reproduction) from what are called the high-level primary drives, such as dominance, fairness, and deference. Drives, such as hunger, encoded implicitly in the motivational system are then used to generate explicit goals, e.g. to find food (Allen and Sun, 2016).

Even systems that do not ascribe to drives often have the machinery to do so, and many implement some drives implicitly. For example, NARS is positioned as a reasoning engine with explicit goals as the only motivation. However, it implicitly includes some drives, similar to the cognitive ones declared by MicroPsi, by virtue of its processing procedures and policies (Wang et al., 2016). CERA-CRANIUM sets a meta-goal of keeping a positive emotional state, which can also be considered a drive[3] (Moreno et al., 2007). Similarly, avoidance of bodily harm embedded in the procedural knowledge of most robotic architectures can be considered a primitive drive of sorts.

Emotions

Broadly defined, emotions are the innate and hardwired mechanisms that perform many functions, such as directing attention, guiding decision-making,

[3]The authors refer to it as personality rather than drive.

and facilitating efficient action in changing circumstances. Although the importance and effects of emotions on cognitive processes are generally not disputed, there are a variety of theories of emotion that diverge on the nature of emotions and mechanisms of their emergence.

The psychological theories of emotion are divided into basic and dimensional. Proponents of the former argue that emotions are composed of a finite set of basic entities, such as anger, fear, joy, etc. The main issue with this categorical representation is that there is no agreement on the basic emotions. Different theories list anywhere from 2 (pain and pleasure) to 24 basic emotions, with most lists containing an overlapping set of 4 to 6 basic emotions (e.g. anger, happiness, sadness, and fear) (Zall and Kangavari, 2022; Cambria et al., 2012). Some of this variation is due to the fuzziness of the emotions themselves, for example, happiness is similar to joy and anger to rage. The dimensional theories of emotions address this issue to some extent and argue that emotional experience is non-specific and often cannot be described as a single emotion (Ortony and Turner, 1990). For example, fear and anger that are part of nearly all lists of basic emotions are themselves strongly correlated and therefore cannot be clearly distinguished (Gray et al., 2001). As an alternative to categorical organization, emotions can be thought of as areas in the space spanned by two axes—valence (positive or negative) and arousal (intensity) (Russell and Barrett, 1999). But, despite its advantages, a 2D model is also not adequate for describing a full range of emotional experiences. For instance, anger, sadness, and disgust are all characterized by negative valence, but cannot be distinguished by arousal alone.

Cognitive architectures that implement emotions are equally divided with respect to definitions. Half of the architectures follow a discrete approach: Kismet lists 8 emotions (including 3 expressive states) (Breazeal, 2003a), SS-RICS—6 (Long, 2017), CERA-CRANIUM—5 (Moreno and de Miguel, 2006), MAMID—4 (Hudlicka and Broekens, 2009), and CoJACK—3 (fear, fatigue, and stress) (Evertsz et al., 2007). In all cases, discrete emotions are defined by their valence and intensity. A dimensional approach is chosen in Sigma (Rosenbloom et al., 2015b), Soar (Marinier III et al., 2009), Clarion (Wilson, 2012), and MicroPsi (Bach, 2012a). Although internally emotions are modal, i.e. represent regions in the emotion space, they can be mapped to the categorical emotions.

Another point of contention is how emotions arise, specifically what stimuli elicit emotions, how valence and intensity of emotions are determined, and what mechanisms and representations are involved. The currently dominant view is that emotions are a result of appraisal (assessment) of external and internal stimuli, which in turn lead to changes in behavior and the state of the agent (Moors, 2014). A number of theories have been proposed that differ in the number of dimensions for appraisal and mechanisms involved (Gratch and Marsella, 2015). As many of these theories are described in computational terms, they are good candidates for inclusion in the architectures. To explain how emotions are implemented in cognitive architectures, we will look into Kismet, MAMID, and MicroPsi.

Kismet (Breazeal, 2003a) combines elements of multiple appraisal theories, such as (Frijda, 1994), (Lazarus, 1994), and (Scherer, 1994). Perceptual information is evaluated with respect to hand-crafted releasers, e.g. desirability of stimulus, speech prosody, threat from stimulus, and goal achievement. The

result of appraisal according to these criteria is a set of somatic markers (tags) that specify arousal (A), valence (V), and stance (S) of the percept (following Damasio, 1994). Emotions in Kismet are defined within a 3D space spanned by arousal, valence, and stance axes. Therefore, A, V, and S contributions from each percept are combined to determine the activations of specific emotions within this space. Finally, a winner-take-all process decides which emotion will take control of the behavior and will be reflected in the facial expression of the robot.

MAMID incorporates elements of multiple appraisal models by Smith and Kirby (2001), Ellsworth and Scherer (2009), and Sloman (2002). The model's affect appraiser module derives the affective state in terms of valence and four basic emotions from external data, internal representation (situation and expectations), and the current goal. Appraisal is done on two levels: low-resolution automatic assessment relies on the universal emotion elicitors, such as novelty and threat, to generate positive or negative affect. High-resolution categorical appraisal uses cognitively complex elicitors to generate a vector of basic emotion intensities. Emotions affect action selection by changing the parameters of processing, as indicated by the empirical data. For example, anxiety and fear reduce attentional and WM capacity, therefore when these emotions are sufficiently activated, decision-making may become impaired (Hudlicka, 2007).

MicroPsi commits to the OCC theory of emotions (Bach, 2012a) but unlike Kismet or MAMID, it captures emotions implicitly and uses a larger number of affective dimensions. In addition to valence and arousal, MicroPsi introduces four more—resolution, selection threshold, goal directedness, and securing rate. These modulators modify access to the memory, perception, action selection, and execution. The values of modulators are determined by the urgency and importance of motives (demands and associated goals) and the ability of the agent to fulfill the tasks that will satisfy them (Bach, 2012b).

Traits

Traits are permanent behavioral characteristics that consistently bias decision-making throughout the lifetime of the organism. Unlike emotions, traits are dispositional, stable, and organized constructs. In the psychological literature, distinction is made between temperament and personality. Although both are trait-like, temperament is believed to be innate, biologically-based, and heritable. Personality is affected by temperament but is much broader and encompasses a variety of individual differences in skills, habits, and social behaviors (Gray et al., 2001). In cognitive architecture research, traits have not received a lot of attention. For example, among the cognitive architectures we reviewed, only four describe personality traits. Out of these, only MAMID natively supports traits.

BB1 is a blackboard architecture that does not natively support personality traits. An instance of the architecture was later used to implement characters for Virtual Theater (Rousseau and Hayes-Roth, 1997). To make the behaviors of characters more realistic, they were imbued with personalities. The implementation is loosely based on the trait and social learning theories. Based on the trait theories, a personality profile has a fixed number of basic traits that are assigned a numeric value (here, in the range $[-10, 10]$). To make personality less rigid, personal experiences can influence expression of traits

in relevant situations. To enable this, weights in the range of $[0, 1]$ and numeric priorities are manually assigned to tie traits to actions for various situations. At runtime, the action that has the highest value (product of trait value and assigned weight) is selected as the most consistent with the current personality profile (Rousseau and Hayes-Roth, 1997).

MAMID likewise follows a parametric route for defining personality but in a more systematic fashion. The model encodes four dimensions of personality traits—extraversion, stability, conscientiousness, and aggressiveness. These traits are in turn mapped onto internal parameters of the model, such as capacity and processing speed of various modules, global time and capacity requirements, threat level, and salience (Hudlicka, 2010). During runtime, these parameters cause small changes in the architecture processing. For example, they can limit the number and types of cues captured by the attention module, or bias selection of goals. As a result, variations in behavior of the agents can be observed (Hudlicka, 2005).

Personality can also be expressed through the parameters governing emotions or drives. For example, in SS-RICS, personality is tied to the parameters that regulate the weight of emotions in the equations for decision-making. By setting permanent values for specific emotions, one can design a personality which is mostly happy or easily surprised (Long et al., 2015). MicroPsi advocates defining personality in terms of the demands (drives), specifically the parameters that govern effects and dynamics of drives (Bach, 2012a). In either case, there is an unresolved issue of mapping these parameters to common personality models, e.g. Big Five (Long, 2017).

7.4 Reasoning about reasoning

Meta-reasoning, according to Flavell's (1979) popular definition, is "thinking about thinking" and encompasses a broad set of abilities that introspectively monitor the internal processes and reason about them (Rhodes, 2019).

7.4.1 Meta-reasoning abilities

Meta abilities are essential for human cognition. It is generally agreed that meta-reasoning performs three main functions: monitoring, control, and learning (Anderson and Oates, 2007; Winne and Azevedo, 2014; Norman et al., 2019; Rhodes, 2019). These functions operate together and enable one another. Monitoring internal state and activities requires analyzing them with respect to the past, present, and future goals. Meta-cognition can be directed at the past and explanatory through retrospective analysis of mistakes. It can also provide an immediate introspective validation of current behavior. This high-level information about validity and applicability of what has been done in the past and is being done currently can be used to regulate the activities and learn new strategies for the future. As a result, meta-cognition can anticipate effects of behavior on expected or desired future goals (Winne and Azevedo, 2014; Cox et al., 2022).

Approximately half of the cognitive architectures have meta-cognitive abilities of some kind. Below, we categorize them using the taxonomy established in the psychological studies:

Monitoring. Self-monitoring is the most basic meta-cognitive mechanism that enables collection and analysis of the data representative of all aspects of the performance and internal state of the system. This includes, for example, maintaining estimates of validity and applicability of knowledge, either via explicit reasoning to remove the inconsistencies and assert epistemic preferences, as in Companion (Friedman and Forbus, 2011) and GLAIR (Shapiro et al., 2007), or in probabilistic terms as in Clarion (Sun et al., 2016) and Soar (Laird, 2012b). Monitoring also applies to keeping track of available computational resources. This is particularly important for maintaining a sustained level of performance in critical situations. Some examples include ICU patient monitoring by AIS-based Guardian (Hayes-Roth et al., 1995) and application of COGNET to an air-traffic control task (Zachary et al., 2000).

Control. Meta-control of execution uses the results of monitoring to align decision-making with the intended goals and available resources. PRS provides a good illustration of the benefits of meta-control in different contexts: improving efficiency of tree search for two-player board games (Russell and Wefald, 1988), determining the amount of deliberation and the utility of replanning during the control of a simulated underwater vehicle (Ogasawara and Russell, 1993), and modifying system goals to respond to detected failures during spacecraft operation (Georgeff and Ingrand, 1989). Another application of self-control is demonstrated in Metacat. Here, self-control relies on the identification of recurring activity from self-monitoring data and breaks the cycle by focusing on a different goal instead of continuing indefinitely (Marshall, 2002).

Learning. Meta-knowledge generated from self-monitoring and past instances of self-control is no different from other declarative and procedural knowledge, except for being self-referential. Therefore, the same learning mechanisms can be used to modify and discover meta-knowledge. Commonly, this is done in a trial-and-error fashion, where the system records its successes and failures and uses them to express probabilistic chances of success of actions or strategies. In ACT-R, from the early versions of the architecture, the utilities of production rules were updated (Bothell, 2017) and a similar mechanism was eventually introduced in Soar (Laird et al., 2012). Reinforcement-like update mechanisms for procedural knowledge can be found in other architectures as well, e.g. SAL (Vinokurov et al., 2013) and FORR (Epstein and Petrovic, 2008). PRODIGY explicitly analyzes the traces of past problem-solving episodes. Meta-information saved for each node in the inference tree (current, failed, successful, and unexplored goals) is used to create new control rules to be applied in similar situations (Borrajo and Veloso, 1994). Meta-knowledge can also be applied to regulate learning itself. For example, FORR monitors its own learning and is able to start anew if progress is insufficient (Epstein and Petrovic, 2008).

7.4.2 Theory of mind

Besides self-monitoring and self-regulatory functions, meta-reasoning is essential for social cognition. Interactions with other agents depend on acknowledging and understanding their mental states and the ability to predict their behavior, which in turn informs one's own decision-making. In the psychological literature, this is known as theory of mind (ToM). Historically, much of

the ToM research focused on the developmental changes that occur in children between 2 and 5 years old. During this period, children gradually form an understanding of the knowledge and beliefs of others (Carlson et al., 2013). To assess their progress, several paradigms were developed, the most common of which is the location false-belief task pioneered by Wimmer and Perner (1983). During the experiment, the subjects are told a story about Maxi, who puts chocolate into a cupboard x. While Maxi is away, his mother moves the chocolate to a different cupboard, y. The subjects are then asked where Maxi will search for the chocolate when he returns. Answering the question correctly (cupboard x) requires an understanding that others may have different beliefs and that those beliefs may be incorrect sometimes.

ACT-R, Soar, Polyscheme, and Sigma are among a few architectures that model aspects of ToM, however neither uses a standard psychological setup. Instead, ACT-R and Polyscheme explore teamwork interaction scenarios. ACT-R models the behavior of a two-person patrol team instructed to proceed to two different stations at the sound of alarm. If both arrive at the same station, one of them should go to another station. Making a decision about where to go without seeing the other person requires a theory of what one would do in a similar situation (Kennedy et al., 2008). A different patrol scenario involves a robot-human team who initially are told to guard the south area of the building but, as they start, the instructions are changed asking them to head west. When the human starts walking south, the robot is expected to realize that the human did not hear the instructions and remind them of the most recent instruction (Trafton et al., 2013). Polyscheme investigates a perspective-taking interaction scenario in which robot and human stand in a room with two traffic cones and multiple occluding elements. At some point, the human gives the robot a command to move toward a cone, but does not specify which of the two cones in the room. In the scenario where the human can see only one cone, the robot is expected to take the human's perspective and use this information to disambiguate the command (Trafton et al., 2005).

Soar and Sigma explore ToM in the context of games. In particular, Soar focuses on adding anticipation to the AI opponent for the death match version of the popular 3D first-person shooter Quake II. Whenever the enemy agent is sufficiently close, the Soar agent assumes the perspective of the enemy and its mental state to predict its action, e.g. heading for health (Laird, 2001). Finally, Sigma focuses on the single-stage simultaneous-move games, such as the Prisoner's dilemma and Stag hunt, and a sequential Ultimatum game (Pynadath et al., 2013). All three, originating in game theory, have been adopted by psychological studies of social interactions (Krueger et al., 2020).

Despite the differences in scenarios and architectural designs, the mechanisms for enabling ToM across these architectures are conceptually similar and all involve mental simulation. ACT-R spawns a submodel with identified beliefs and goals of the other agent and uses this submodel to simulate the other agent's decisions (Trafton et al., 2013). Since running an additional instance can be costly, other architectures develop leaner solutions. Soar uses a specially constructed set of rules to represent the other agent's knowledge and decisions (Laird, 2001). Similarly, Polyscheme has dedicated schemas for this purpose (Trafton et al., 2005). Finally, in Sigma, the independent decision-making of the other agent is represented as an additional dimension within its graphical representation to minimize the computation (Pynadath et al., 2013).

7.5 Summary

- Some form of reasoning or decision-making is involved in most cognitive functions, from perception to logical problem-solving. Reasoning is often further subdivided into theoretical, which operates on and results in beliefs, and practical, which proceeds from beliefs to intentions and actions. However, this separation is somewhat artificial since theoretical reasoning is often a part of deciding to act and mechanisms used by both are conceptually similar.
- Although logic is often associated with theoretical reasoning, everyday human reasoning is not always logical, rational, or optimal. Instead, it relies on potentially incomplete and uncertain premises, which can be modified or nullified by new information. Therefore, the focus of most cognitive architectures is on the non-monotonic approximate forms of reasoning, such as induction, abduction, and analogy.
- Practical reasoning or decision-making is about choosing the best possible response for the given circumstances, and must balance utility and timing. Because it has the most practical application, decision-making is well-represented in cognitive architectures. In addition, it is supplemented with biologically and psychologically inspired details, such as behavior modulation via emotions, drives, and permanent traits.
- Meta-reasoning is the most recent addition to cognitive architectures and is currently the least explored. Typical mechanisms include self-monitoring by gathering information about the current state of the system, and self-regulation by applying this knowledge to improve decision-making and learning.

8 Putting It All Together

In the previous chapters, we discussed individual components of cognitive architectures. But the entire system is more than the sum of its components—the same module will perform differently depending on the type and number of connections to other elements of the system. Here, we focus on the bigger picture of how these elements fit together in different architectures.

Section 8.1 describes a generic cognitive cycle and the flow of information between the typical components.
Section 8.2 identifies several common classes of the architectural topologies, their variations, and properties.
Section 8.3 discusses timing of cognitive cycles based on psychological motivations and practical needs.

8.1 Cognitive cycle

All cognitive architectures we reviewed execute a perception-action cycle. During each iteration, sensory input is received, aggregated, and processed to output an action. Once the action is performed, the next cycle starts. This circular process is a highly abstracted version of the human cognitive cycle (Fuster, 2017).

Figure 8.1 shows a schematic representation of a typical cycle found in virtually all architectures. Boxes in the diagram correspond to basic components discussed in the previous sections: perception, transient and long-term memory storage, and behavior generation, which combines reasoning and decision-making.

The following is a high-level summary of the processing within the components and the flow of information along the arrows connecting them:

Perception ↔ STM. Sensory information processed in the perception module enters the short-term memory (STM). STM may contain items from the previous cycle, some of which will be updated or discarded. Exchange of information between perception and STM is modulated by attention in both directions. The bottom-up attention mechanisms increase the chance of salient or anomalous stimuli moving to the STM storage. In the opposite direction, top-down attention influences selection of percepts that better match the contents of the temporary memory.

STM ↔ Behavior generation. The perceptual module provides information about the current state of the world and the state of the agent (through proprioception). STM may also contain the set of current goals and beliefs. Based on this information, behavior generation module(s) produces new beliefs, goals, or motor commands.

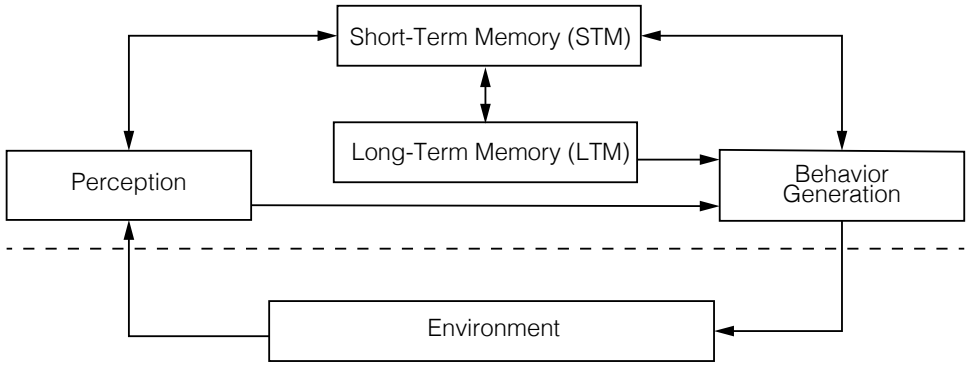

Fig. 8.1 Illustration of the modules and the flow of information between them for a typical cognitive architecture. Solid arrows indicate inputs and outputs.

STM ↔ LTM. The items stored in STM are often used to pull the relevant information from long-term memory (LTM). This information can be potentially of any kind: conceptual knowledge to enrich the context in STM, procedural knowledge needed to satisfy the goals, related past experiences, or even spontaneously recalled and irrelevant items. In the opposite direction, the contents of STM may be permanently stored or used to update the information in LTM as a part of learning.

Perception → Behavior generation. External stimuli or proprioception can trigger reflexive responses, such as braking responses or balancing, that bypass deliberate decision-making.

8.2 Topology and processing

The perception-action cycle is similar across all cognitive architectures only on a high level; there are many differences in how the computation is organized within each component and in the information flow between them. In the previous chapters, we already discussed common knowledge representations and the associated processing and inference options for different components of cognitive architectures. Here, we consider various approaches to organizing computation within the system and its runtime operation. To do so, we distinguish four typical topologies—layered, distributed, centralized, and hierarchical—as illustrated in Figure 8.2. For each type, runtime operation will be evaluated with respect to three dimensions: parallel/serial, synchronous/asynchronous, and distributed/centralized.

Layered. Layered design helps isolate modules with different states, computational complexity, and running times. Specifically, reactive control is separated from costly planning. Reactive control connects the sensors to actuators and is fast because it relies only on the sensor information, is virtually stateless, and focused on the present. Deliberative planning has the opposite characteristics: it needs a world model and is concerned with the future actions leading to the goal. They communicate through a scheduling layer between them. The scheduler receives input from and dispatches appropriate primitive operations

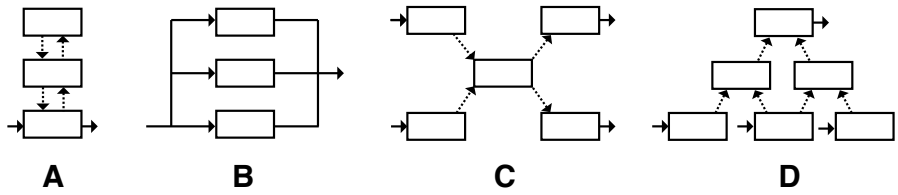

Fig. 8.2 Schematic illustrations of the most common topologies: A) layered, B) distributed, C) centralized, D) hierarchical.

to the reactive control. It also monitors planning progress and may terminate it or request more depending on the current situation and task (Gat, 1998).

Robotic architectures 3T, ATLANTIS, Saphira, ERE, and GLAIR are typical examples of layered architectures, where a fast reactive control and a slow deliberative planner are connected with a sequencing mechanism. These types of systems combine asynchronous and parallel execution of the reactive and planning modules with serial centralized control performed by the sequencer.

This design is motivated mainly by empirical observations and real-time performance targets, but has parallels with psychological phenomena; primitive behaviors within the reactive layer are similar to reflexes and Suchman's (1987) situated actions in that they are prompted by external events and involve little to no processing (Hexmoor et al., 1992; Gat, 1998).

Distributed. Distributed organization emphasizes parallel and asynchronous execution for enabling reactive and emergent behaviors. The classic example is the Subsumption architecture, where a number of loosely coupled concurrent processes implemented as finite state machines run each at their own pace while communicating through predefined inputs and outputs. Although all processes produce outputs, only some of them reach actuators and the rest are suppressed (or subsumed) (Brooks, 1986). Besides Subsumption and behavior networks (Maes, 1991), there are no architectures implementing purely distributed, asynchronous, and memoryless processing. In some instances, this approach is applied within individual modules, e.g. executor submodules in RCS (Albus, 1997) and saccade control in IMA (Kawamura et al., 2008).

Centralized. Centralized models are the most common. Prototypical examples are the blackboard architectures and production systems, in which a number of modules run independently and communicate through a shared data structure. BB1 is one of the early blackboard architectures, where specialized knowledge sources (e.g. perception, reasoning, and action) run in parallel and asynchronously. All coordination and communication happens through posting and inspecting the items recorded in the global data structure called the blackboard. Changes to the blackboard in turn trigger execution of other knowledge sources (Hayes-Roth et al., 1989). Although it is theoretically possible for all knowledge sources to operate completely in parallel, most implementations have a scheduler which selects a single knowledge source at a time. The selected knowledge source and information relevant to it (context) are referred to as the focus of attention (Pfleger and Hayes-Roth, 1997).

There are many variations on the blackboard architecture in different cognitive architectures. In COGNET, the blackboard is organized hierarchically according to the levels of abstraction from basic sensor events to the objects

identified in the scene (Zachary et al., 1990). Copycat implements more fine-grained parallelism, since its knowledge-source-like codelets contain only a few lines of code, and uses a semantic network as a representation for its blackboard. During each cycle, a single codelet is chosen to run, which activates certain elements in the blackboard (Hofstadter, 1994). PRS introduces the following enhancements: interruptible knowledge sources, procedural representation of information in the blackboard instead of a rule-based one, intention graph instead of the serial scheduler for managing complex priority mechanisms, and meta-level knowledge sources that can modify the intention graph layout dynamically at runtime (Ingrand and Coutance, 1993). Lastly, DUAL removes the separation between the knowledge sources and the blackboard; the knowledge sources themselves are the blackboard and have access only to the parts of the context exposed through the connections to the neighbors (Kokinov, 1994a).

More recent instances of blackboards depart further from the original concept. In CORTEX, a central shared blackboard is implemented as a complex graph-based data structure, to which all agents, both deliberative and reactive, have access, thus allowing it to represent the current state and future actions (Marfil et al., 2019). Polyscheme minimizes the blackboard to only store the results of computation of multiple knowledge sources (called specialists), each of which has its own memory and an arbitrary representation. The focus manager then considers suggestions from all specialists, chooses one as the next focus of attention, and broadcasts it to the rest. Since the specialists are not limited in their structure and implementation, all communication between them is done through a common intermediate representation called interlingua (Cassimatis, 2005). LIDA adapts the elements of the original blackboard system and Copycat's fine-grained approach to parallelism for implementing the Global Workspace Theory. LIDA's blackboard is accessible to a large number of small codeletes that concurrently read and write to it, while competing for attention, which (similarly to Polyscheme) broadcasts the winner's context to the rest of the codelets (Faghihi et al., 2012). The blackboard in ARCADIA is a highly volatile structure with unlimited capacity. The knowledge sources with their own memories and internal representations translate results into common interlingua (as in Polyscheme) and place them in the blackboard. At the end of each cycle, one element is broadcast for all knowledge sources to work on next (Bridewell and Bello, 2016).

Production systems are closely related to blackboard systems and have a similar high-level structure characterized mainly by production rules (procedural memory) and a dynamic workspace referred to as working memory (WM) (see Section 2.2). The main difference lies in the representation: production rules are fine-grained and uniform if-then-else condition rules, whereas knowledge sources are more arbitrary both in implementation and scale.

Similarly to blackboard systems, there are many variations to the basic production system design. For example, WM in ACT-R is represented by a set of buffers that hold only one element. For every cycle, only a single production among those matching the contents of the buffers is selected to fire (Anderson and Lebiere, 2003). EPIC, like ACT-R, has a number of dedicated buffers for different types of information (e.g. visual and auditory percepts, goals) forming its WM. However, EPIC allows all rules that match the contents of the WM to execute in parallel, eliminating the serial control bottleneck

(Kieras, 2007). CAPS separates not only the WM stores for different types of information, but also groups the production rules into separate modules (e.g. spatial, verbal, quantitative). The rules within each module can fire in parallel and update the respective WM (Just and Varma, 2007). Soar, unlike most other rule-based systems, uses rules to support the selection, application, and termination of operators, which describe primitive and abstract actions. When abstract actions are proposed by rules, they become goals and are dynamically decomposed into sequences of operators, that are then selected by more rules (Laird et al., 1998). Soar's WM contains goals, percepts, and retrievals from the LTM, etc. (Laird, 2012b).

Hierarchical. In a hierarchical design, modules are separated into layers, with the layers below subordinate and feeding input into the layers above. A classic example of such organization is RCS, whose modules (nodes) are organized in levels defined by temporal and spatial resolution, i.e. low-level sensors and actions at the bottom and long-term planning on top (Albus, 1991). Similar design is adopted in architectures that mimic the hierarchical organization of the human visual cortex, e.g. DeSTIN (Goertzel, 2012a) and Leabra (O'Reilly et al., 2016).

Centralized blackboard-like design is clearly the most common topology among our broad selection of cognitive architectures. Although blackboards were proposed more than fifty years ago, they remain a popular choice. Several factors likely contributed to their longevity. For one, the concept of a blackboard is intuitive, flexible, and easy to implement, making it an attractive option for experimentation. Blackboards work well not only for applied research, as demonstrated by a number of successful applications (e.g. BB1, CORTEX, LIDA), but are useful for studying human cognition. In fact, various production systems, such as ACT-R, Soar, CAPS, and EPIC, have been used to replicate human performance on a range of tasks. Furthermore, there is evidence suggesting that blackboard-like functions can be a useful abstraction of perception and decision-making in the brain (Roelfsema and de Lange, 2016; Worden et al., 2021). However, without comparative evaluations against other approaches, the superiority of blackboards cannot be established (more on evaluation in Chapter 10). Nevertheless, convergence of both engineering and psychologically motivated research on this solution makes it a promising candidate for further exploration.

8.3 Timing

Reactivity is one of the key components of intelligence and an important factor to consider when designing cognitive architectures (according to desiderata in Chapter 1). Biological and artificial organisms alike must respond in time to the changes in the environment in order to survive and function. This is widely recognized in cognitive architectures, but implemented in different ways and with different purposes.

More application-oriented architectures, particularly those that are physically embodied, require speed of processing that is commensurate with external events to avoid harm and delays. While real-time operation is less important for simulating behavioral and neural phenomena, it is still necessary to show that computations can at least in principle satisfy biological constraints.

Table 8.1 Cognitive cycle durations reported for the cognitive architectures. The "processing" column indicates what type of processing is included in the cycle: P—perception or R—reasoning, i.e. all processing beyond perception. For some cognitive architectures, the range of values for perception and reasoning is provided, in which case they are reported with a "+" sign. "Real-time" means that the exact range is not available, but the cycle completes in time commensurate with the rate of the sensors or events.

Architecture	Cycle duration	P	R	References
3T	10–1,000 ms		+	(Bonasso and Kortenkamp, 1995)
ACT-R	85 ms + 50-70 ms	+	+	(Anderson et al., 2004)
ATLANTIS	10-1,000 ms	+	+	(Gat, 1993)
BBD	100–250 ms	+	+	(Reeke et al., 1990)
CAPS	25–100 ms		+	(Just and Varma, 2007)
CARACaS	100 ms		+	(Hansen et al., 2006)
CHREST	200 ms + 100 ms x #items	+		(Smith et al., 2009)
DIARC	80–90 ms		+	(Joshi et al., 2012)
EPIC	50 ms		+	(Meyer and Kieras, 1994)
IMA	100 ms	+		(Kawamura et al., 1993)
Kismet	33 ms	+	+	(Breazeal et al., 2001)
Leabra	100 ms	+	+	(Kachergis et al., 2014)
LIDA	200–500 ms	+	+	(Franklin et al., 2016)
MHP	50–200 ms + 25-170 ms	+	+	(Card, 1981)
MIDAS	100–200 ms	+		(Corker and Smith, 1993)
PRS	20–500 ms	+	+	(Huang et al., 1996)
SASE	30–100 ms	+	+	(Weng, 2007)
Sigma	250 ms		+	(Rosenbloom et al., 2015a)
Soar	1–50 ms		+	(Laird et al., 2011)
SPA	50 ms		+	(Stewart and Eliasmith, 2009)
STAR	25 ms	+	+	(Kotseruba, 2016)
Subsumption	10 ms	+	+	(Brooks, 2014)
TCA	10–3,000 ms	+	+	(Simmons et al., 1992)
Ymir	100–2,000 ms	+	+	(Thórisson, 2002)
BB1	Real-time		+	(Hayes-Roth, 1990)
BECCA	Real-time	+		(Rohrer, 2007a)
CIRCA	Real-time		+	(Atkins et al., 1997)
COGNET	Real-time		+	(Zachary et al., 2000)
CORTEX	Real-time	+	+	(Mendoza et al., 2018)
DAC	Real-time	+	+	(Vouloutsi et al., 2015)
ERE	Real-time		+	(Drummond et al., 1991)
FORR	Real-time		+	(Epstein, 1994)
NARS	Real-time		+	(Wang, 2009)
RALPH	Real-time		+	(Russell and Wefald, 1989)

A summary of cognitive cycles durations is given in Table 8.1. Reported values usually include decision-making or perception and decision-making together, however, not motor actions because of their high variability. Note that for nearly half of cognitive architectures (not shown in the table), the duration of cognitive cycle is not specified.

8.3.1 Matching human response time

Behavioral constraints for cognitive cycle duration are obtained by measuring a range of human response times for specific tasks. A model that can replicate these timings can be used to 1) generate human-like behavior in the same or different conditions, 2) estimate the effects of changes in the task or the characteristics of the operator, and 3) help revise the designs of the tasks or the environments where they are performed.

Matching human response times is particularly common in human performance modeling. A well-known example is the Model Human Processor (MHP), an early model of human information processing designed for practical

purposes but grounded in cognitive science. MHP is a family of parametric models developed for simple procedures, such as reading text, reaching for a button, or binary choice with the purpose of analyzing and improving basic cognitive tasks (Card, 1981).

Human processing times have been determined for other domains and more difficult tasks. For example, COGNET models behaviors of the air traffic control operators and defines specific timings depending on the tasks performed and different levels of cognitive load (Zachary et al., 2000). MIDAS matches the performance of the helicopter pilot and then uses the parameters to evaluate and improve the cockpit design (Corker and Smith, 1993). CHREST investigates human perception and memory in tasks, such as playing chess, to match human response times and explain the internal mechanisms of the behavior. Parameters for detecting the chess piece (200 ms), recognizing it (100 ms/item), and moving it to a different spot on the board (50–100 ms) were extracted from the analysis of the human experiments (Smith et al., 2009).

Several cognitive architectures, particularly production systems, such as ACT-R, EPIC, and CAPS, seek universal parameters for basic tasks and all converged on the mean cycle duration of about 50 ms. This time usually refers only to the duration of the decision cycle. In ACT-R, this involves recognizing patterns in buffers, firing a production rule, and updating the buffers for the next cycle, which is likened to the flow of information from cortex to the basal ganglia and back. Perception may take longer and often depends on the implementation. For example, in ACT-R 5.0, the cost for object identification has been reduced from 185 ms to 85 ms (Anderson et al., 2004). Motor actions are also not included in the cycle. CAPS aims at a geometric mean of the cycle around 50 ms, however durations can vary depending on the implementation. For example, cycles for reading comprehension and mental rotation models were 60 ms and 30 ms, but a model for solving the Tower of London took 556 ms per cycle (Just and Varma, 2007).

EPIC also allocates 50 ms to updating the memory and firing rules (Kieras et al., 1998). An additional 50–250 ms are allocated for perception (depending on the modality and the perceptual task), and 60 ms for initiating and performing a motor action (Kieras and Meyer, 1994). The timing of 50 ms per cycle in EPIC is linked to the periodicity of EEG brain activity (α rhythms) and the distributions of reaction times for the given tasks. Since α rhythm is strongly correlated with age, modifying the cycle duration reflects the changes in performance between older and younger participant groups (Meyer and Kieras, 1994). Similar effects have been observed from modifying the cognitive cycle of ACT-R (Salvucci et al., 2004).

The LIDA architecture reports a cycle duration between 200–500 ms, including perception, attention, and action selection. This timing is related to theta, gamma, and alpha oscillations (Franklin et al., 2016). Theta and gamma oscillations a believed to be important for synchronizing and communicating between different brain areas (Lisman and Jensen, 2013), functionally similar to the periodic global broadcasts in LIDA.

Psychologically motivated cognitive architectures correlate their structure and performance with neural phenomena, but do not model them directly. For example, Leabra, BBD, and SPA consider processing on the level of individual neurons and neuron populations. Leabra derives its 100 ms limit for prediction and sensation phases from the α rhythm (Kachergis et al., 2014).

Since each BBD is unique, reported cycles vary from 100 to 250 ms, which includes the sensory input processing, computing the states of all neuronal units, and generation of motor commands (Reeke et al., 1990; McKinstry et al., 2008). These durations are often associated with oscillations in the visual cortex and in the cerebellum (a brain area associated with motor control and coordination).

Although there are observed correlations between brain oscillations and behavioral performance, the exact neural mechanisms linking the two are largely unknown (Ruzzoli et al., 2019). SPA attempts to provide a neurological basis for the 50 ms cycle found in many psychological cognitive architectures by modeling basal ganglia with well-established properties and the time constraints of neuron membranes and GABA neurotransmitter (Stewart and Eliasmith, 2009).

Overall, reported cognitive cycles fall within the range of 10 ms to 10 s, defined by Newell (1990) as the cognitive band and which also happens to be an appropriate horizon for most reflexive and deliberate actions.

8.3.2 Matching external events

In most real-world applications, reacting late is tantamount to not reacting at all. Reaction time is determined by many factors, both internal (the design of the agent and available computational resources) and external (the task and the environment). Thus, appropriate reaction times can vary significantly, from milliseconds to hours. Below, we discuss a number of techniques developed for dealing with time and resource restrictions.

Fixed cycle duration. The simplest approach is to fit all processing (from perception to dispatching motor commands) or parts of it within a set time interval, matching either the frequency of the expected external events or the temporal resolution of the sensors. For example, 25 ms cycle of the STAR model for playing video games matches the frame rate of the game screen (Kotseruba, 2016). Cognitive architectures designed to interact with humans also focus on quick perception and motor response. For example, IMA implements face tracking at 10–12 Hz to detect sudden movements of the user (Kawamura et al., 1993) and Kismet aims at 33 ms visual processing and face movements matching its sound production (Breazeal et al., 2001).

Bounded cycle duration. A less restrictive approach sets an upper bound for the response times, at least for some operations. Typically, the bounds are placed on deliberation, assuming that hard deadlines are available. For example, PRS limits the number of options open to deliberation, uses decision-making mechanisms that have strict execution time bounds, and prevents the system from reconsidering the plans as long as they do not fail (Ingrand and Georgeff, 1990). CIRCA likewise takes into account deadlines when choosing actions, and in addition, precomputes a set of plans for the most probable failure cases (Atkins et al., 1997).

Anytime response. A disadvantage of both fixed and bounded processing cycles is that the system may commit too early to a suboptimal action even if more time for deliberation is available, use outdated information about the environment, or react too late. The anytime algorithms solve this problem by a) having an answer available at any point in the execution and b) improving

its quality, if more time is available (Dean and Boddy, 1988). Typically, this is applied to decision-making (NARS), action selection (ERE, BB1, CORTEX, IMA), and motor control (Ymir).[1]

Decoupling cycles. Naturally, some steps in the processing pipeline take longer than the others. Perception (especially vision) and deliberation can take considerable amounts of time. On the other hand, dispatching motor commands (after the decision has been made or as a reflexive response) can be done very efficiently. Thus, forcing all processing into a fixed cycle duration can compromise the quality of more time-consuming steps and negatively affect the quality of response. Separating the operations based on their expected completion times and organizing them into separate feedback loops is a common approach, pioneered in the "reactive planning" architectures (e.g. ATLANTIS, 3T, ERE) (Bresina and Drummond, 1990; Gat, 1993; Bonasso and Kortenkamp, 1995). Along similar lines, RCS is hierarchically organized according to computational complexity and time/space planning horizons, from low-level motor control on a millisecond scale to long-term strategy forming over days and weeks (Huang et al., 1996). Ymir runs multiple feedback loops at different frequencies for motor actions (100–500 ms), process control (0.5–2 s), and topic interpretation (1 s) (Thórisson, 2002).

8.4 Summary

- Nearly all cognitive architectures implement a similar perception-action cognitive cycle with the same basic functional components.
- The most common architecture topology is the multi-agent structure with shared memory, which includes the blackboard and production systems. Together with variations, such systems comprise more than half of all cognitive architectures.
- The cognitive cycle of most cognitive architectures falls within Newell's cognitive band (between 10 ms and 10 s). This is sufficient for prompt reactions to the external events, matches human reaction times for simple tasks, and has correlations to the cortex-wide oscillations that coordinate processing across multiple brain areas.
- There are numerous techniques for achieving real-time execution—from fixed and bounded processing time to more flexible anytime approaches, however, no single technique is sufficient. Therefore, the majority of cognitive architectures combine them, e.g. bounded perception, decoupling of computationally expensive planning from motor control, etc.

[1] Anytime perception is uncommon but also possible

Part III

WHAT CAN COGNITIVE ARCHITECTURES DO?

The following two chapters are about what cognitive architectures can do and how well they do it. Chapter 9 starts with assessment of the core cognitive abilities demonstrated by cognitive architectures in practice. It then examines the applications of cognitive architectures by dividing them into four categories—from abstract psychological tasks to commercial applications. Chapter 10 describes approaches to evaluating performance of cognitive architectures on the level of task and ability, in qualitative and quantitative terms.

9 Practical Applications of Cognitive Architectures

The field of cognitive architectures positions itself as research-oriented. This intent is clear in the numerous proposals and desiderata that emphasize understanding and modeling human cognition over the utility in real-world applications. However, successful replication of the human mind's internal processes should in principle result in a similarly human-like outward behavior. Thus, even as primarily research tools, cognitive architectures should be expected to act in a useful and non-trivial manner in various domains. In this chapter, we discuss the achievements of cognitive architectures in terms of the concrete tasks they can perform.

Section 9.1 briefly introduces the categorization of tasks into four groups. The following sections consider these groups and list the representative tasks in each.
Section 9.2 discusses the tasks performed in the context of psychological experiments. They are referred to as abstract, since they are usually highly abstracted versions of everyday tasks.
Section 9.3 looks at the tasks that involve perception and reasoning, such as categorization and solving puzzles.
Section 9.4 gives an overview of the tasks that require decision-making and motor control, from basic navigation and obstacle avoidance to complex tasks involving multiple steps.
Section 9.5 discusses the tasks that involve interaction with animate or inanimate agents, such as social and robotic assistants.
Section 9.6 covers several notable uses of cognitive architectures for solving real-world problems and in commercial applications.

9.1 Overview of tasks and applications

One major challenge in quantifying tasks and applications lies is the absence of coherent and widely accepted taxonomy. The crux of the matter is defining what constitutes a "task" (Companion and Corso, 1982). In everyday language, task is understood as any activity that a person can do and that can be denoted with a verb, but such a definition is too broad to be useful. Therefore, various categorization schemes have been devised. In the human factors research, tasks have been categorized by the properties of the environment (e.g. natural or artificial), the operator (e.g. physical and mental state), the task itself (e.g. devices, stimuli, subtasks), and the organization (e.g. culture and economics) (Gawron et al., 1991). Artificial intelligence (AI) research focused on the visually recognizable activities, which were organized hierarchically from simple limb movements to semantically meaningful actions, such as gestures,

behaviors, group interactions, and events (Beddiar et al., 2020). Both of these taxonomies do not fit our purposes. While human factors taxonomies are extensive and potentially cover any conceivable activity, they are cumbersome to apply, particularly given the dearth of information in the sources available for cognitive architectures. However, the visual activity categorization is too narrow, as it does not capture the internal cognitive processes that the cognitive architectures tend to focus on and clumps together many complex actions.

An alternative approach is to derive categories from data. In our past work, which was one of the first attempts to tally the abilities and applications of cognitive architectures, we clustered applications found in papers as robotics, psychological experiments, games and puzzles, and natural language processing (Kotseruba and Tsotsos, 2020). Here, we instead distinguish the following five broadly defined groups:

Abstract tasks. For lack of a better term, the category of abstract tasks includes the systematically designed and controlled activities that have long been fundamental in psychology research.
Perception and reasoning tasks. These tasks form a basis for many applications that require sensing stimuli and manipulating information, for example, categorization and problem-solving. However, a number of cognitive architectures perform these as standalone applications.
Procedural tasks. Procedural tasks go beyond forming a belief or solving a problem and instead result in an intent to act or an execution of a physical motion, e.g. controlling a robot, physical or simulated.
Interactive and social tasks. Interacting with others combines elements of perception and problem-solving abilities with decision-making and motor control. The applications in this group often involve assistive functions, following commands, or cooperating to achieve common goals.
Real-world and commercial applications. We consider separately the applications of cognitive architectures to solving real-world problems in space, manufacturing, and assistive robotics.

The last three groups in this categorization correspond to Bennett's (1971) tri-factor hypothesis that separates human tasks relating to ideas, things, and people. Some of these tasks are also contained in the abstract group but in a highly simplified and reduced form. In terms of implementation, abstract tasks are generally less demanding than interactive and real-world ones, which put stricter constraints on the responsiveness and output quality.

This chapter is based on a manually annotated corpus of over 2,800 papers. We considered only those applications for which there was evidence of implementation, such as the authors' statement to that effect, evaluation, demo, a link to the code, etc. Over a 1,000 individual implementations were found, including some intended for use in the manufacturing and safety-critical domains, as well as many proof-of-concept demonstrations. In absolute terms, the applications in the first three groups comprise more than 80% of the total, as shown in Figure 9.1A.

Within each group, there are several clearly dominant architectures. Most notably, more than half of all abstract tasks are simulated in ACT-R and more than one-third of all perception-related tasks by ART. Social and real-world tasks are relatively uncommon and there are no clearly dominant architectures

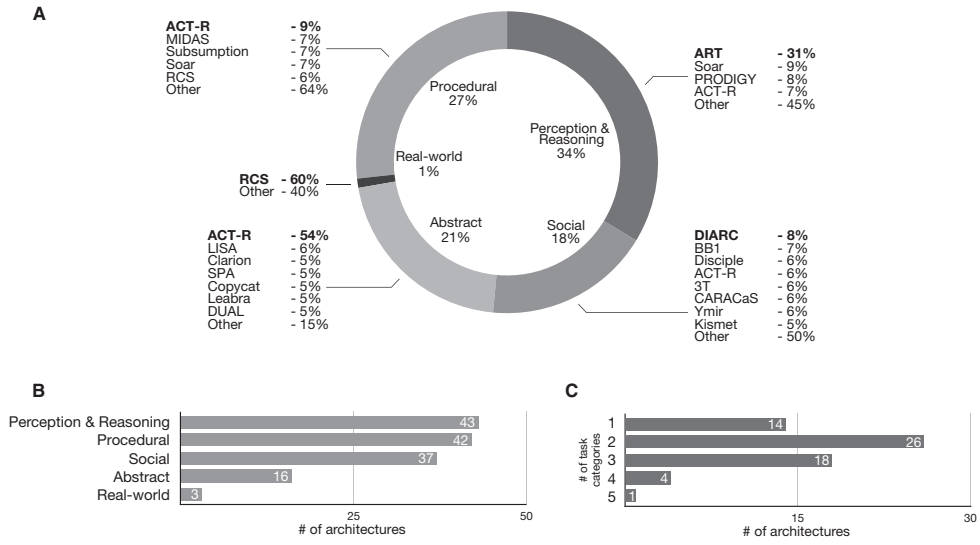

Fig. 9.1 Plots illustrating different aspects of practical abilities of cognitive architectures. A) Distribution of tasks in each category and cognitive architectures with the largest share in each category (percentages normalized by the category). B) Number of cognitive architectures implementing each category of tasks. C) Number of cognitive architectures that implement tasks from different categories.

in this category; about half of all applications are represented by eight architectures, either on a robotic platform (DIARC, Ymir, CARACaS, 3T, Kismet) or in software only (ACT-R, Disciple, BB1).

Figure 9.1(B and C) provides a breakdown of the categories of tasks performed by individual architectures. In terms of diversity, only about one-third of cognitive architectures cover three or more categories of the tasks. The tasks in different categories are also not distributed evenly, with perception/reasoning, procedural, and social tasks being more common categories.

Each of the task categories consists of a large number of varied tasks. The next sections will help better understand the specific abilities of cognitive architecture by discussing representative examples of the tasks from each category.

9.2 Abstract tasks

As mentioned in the previous section, abstract tasks are commonly performed in the context of psychological experiments. Such studies isolate particular aspects of cognition (encoded as dependent variables) and measure responses of human subjects while experimental conditions (independent variables) are being manipulated. The goal is to show that the dependent variable associated with the cognitive phenomenon of interest is affected by the changing conditions in a statistically significant way, i.e. rejecting the null hypothesis.

To reduce potentially confounding factors and to control experimental conditions, tasks are intentionally simplified and often have little to do with everyday activities. For example, the subjects may be asked to recall lists of unfamiliar symbols presented visually or audibly, solve algebraic or geometric

problems (sometimes specified in non-standard notation), or perform visual search in the images depicting the symbols with different orientations and colors.

The artificial and abstract nature of these "nonsense" tasks (Miller, 1967) has cast doubt on whether they can truly represent the intricacy of human cognition and generalize to the real-world conditions (Holleman et al., 2020). But in the case of cognitive architectures, implementations do not always capture all aspects of the task without simplifications and omissions. Thus, even if the task performed by the human subjects is real and complex, the implementation may reflect only a narrow facet of the observed performance. Nevertheless, abstract tasks are quite common, mostly as a way to quantify the similarity to human behavior (as discussed in Chapter 10 on evaluation). Below, we list some representative examples.

9.2.1 Perception and attention

As was mentioned before, vision is the most represented and studied perceptual modality in cognitive architectures. Thus, the majority of perceptual tasks involve vision and visual attention.

One of the most common visual tasks is visual search, i.e. using sight to find objects with certain characteristics among clutter. In everyday life, it would be searching for a particular icon on the phone screen or looking for a friend in the crowd. However, typical experimental setups for this task are much simpler. Usually, the subjects are asked to find a target among similar looking objects (distractors). Both the target and distractors are simple shapes, such as rectangles, circles or triangles, with different basic characteristics, such as color, orientation, or size. Several cognitive architectures have been tested on this type of task, including ACT-R (Nyamsuren and Taatgen, 2013b), DIARC (Scheutz et al., 2014), and EPIC (Kieras et al., 2015).

Other attention-related tasks have also been modeled in cognitive architectures: a computational explanation for attentional blink in LIDA (Madl and Franklin, 2012), change blindness simulated in ARCADIA (Bridewell and Bello, 2015), a Leabra-based neural model of the Stroop task (Herd et al., 2006), and an EPIC model of auditory attention demonstrating a well-known cocktail party effect (Kieras et al., 2016).

9.2.2 Memory

Another large group of tasks modeled in cognitive architectures involves working memory. A classic experimental paradigm tests human abilities of recalling items after brief presentation. Since this is quite difficult for humans but trivial to implement on a computer, well-performing models are not judged by how perfectly they reproduce presented stimuli, but rather by how well they emulate the relative times for different conditions and the types of errors the human subjects make. For example, after being presented with sequences of stimuli, recall accuracy depends on the number and complexity of the items, how long they were presented and in what order, although there are significant individual differences. These aspects have been investigated in ACT-R (**?**; Daily et al., 2001; Chuderski et al., 2006; Schneider and Anderson, 2011; Peebles and Jones, 2014), EPIC (Kieras et al., 1998), and CHREST (Gobet,

1998). Another common task for measuring working memory capacity is the n-back task, where a person is presented with a series of stimuli and must decide whether the current stimulus matches one from n steps before. The larger the number n, the more difficult the task. Neural mechanisms of this task were investigated in the Leabra (Chatham et al., 2011) and SPA (Gosmann and Eliasmith, 2015) models.

Functioning of episodic memory was also investigated in several architectures. Since there are fewer architectures implementing this kind of memory, and it is less studied than working memory, the tasks bear little relation to one another. For example, the SHRUTI model explained the formation and recall of simple episodic memories and simulated their degradation from neurological diseases (Shastri, 2001). The DUAL architecture was used to simulate the effects of familiarity and typicality of the episode on its retrieval (Kokinov et al., 2007; Grinberg and Kokinov, 2019).

9.2.3 Reasoning

Reasoning is represented by a broad range of symbol manipulation tasks in various contexts. Two common examples are categorization and analogical inference. Categorization involves classification of objects based on various attributes and has been demonstrated by ACT-R (Dickison and Taatgen, 2007), Leabra (Rougier and O'Reilly, 2002), and SPA (Schröder et al., 2014), while visual and verbal analogical inference has been investigated in LISA (Morrison et al., 2004), DUAL (Kokinov, 1990), Clarion (Licato et al., 2014), and Copycat (Hofstadter, 1994). Other aspects of reasoning include mechanisms for solving simple mathematical problems ACT-R (Ritter et al., 2002), use of mental imagery for solving geometric problems (Soar; Lathrop and Laird, 2007) and performing mental rotation of shapes (CAPS; Just and Carpenter, 1985).

9.2.4 Learning

Acquisition and loss of knowledge has been examined across multiple experiments. There are models of learning new procedural skills (e.g. using a different type of mouse for data entry modeled in ACT-R; Kim et al., 2007), as well as learning declarative knowledge, such as recognition of anomalies in images (ACT-R; Kennedy and Patterson, 2012), and learning from the previous experience which deck to take the next card from in the Iowa Gambling task (ACT-R; Fum and Stocco, 2004).

9.2.5 Multitasking

People often perform two or more tasks simultaneously. Some combinations of tasks are relatively easy, like talking to a passenger while driving a car, and some are more difficult, for example, studying while watching a movie. As with the memory tasks, programming a computer to perform multiple activities is easy. But the goal of most models is to mimic typical error patterns and degradation of performance during concurrent tasks. For example, an ACT-R model of multitasking correctly modeled increased errors in a dual-task setup consisting of n-back and tone counting tasks (Borst et al., 2015). An important element of multitasking is task switching, i.e. shifting between different tasks. Switching causes time delays and higher error rates, even if the tasks are very

simple. Another ACT-R model replicated this effect on a dual task involving counting and categorizing characters (Altmann and Gray, 2000).

9.3 Perception and reasoning tasks

The human brain excels at extracting and analyzing information from noisy sensory data. Many tasks depend on this ability, but it has been notoriously difficult to implement with symbolic algorithms. Some examples of the tasks in this category are classification, pattern recognition, and clustering that historically have been more amenable to subsymbolic representations and statistical methods. Not surprisingly, the architectures with significant subsymbolic component are applied to these types of tasks more frequently. One architecture in particular, ART, has performed the largest number of tasks in this category owing both to its specialization in perceptual processing and the availability of easy to use models. Reasoning tasks, such as game playing, problem-solving, and language understanding, are more naturally handled by largely symbolic architectures, such as Soar, PRODIGY, ACT-R, and FORR.

9.3.1 Perceptual processing

As vision is the dominant sensory modality, the majority of applications are related to vision as well. However, most of them are fairly simple, such as classifying small sets of abstract patterns (ART; Carpenter and Grossberg, 1987a), distorted letters of the alphabet (CogPrime, Arel et al., 2009; ART, Kane and Paquin, 1993; CHREST, Lane et al., 2019), or wafer bin maps with various defect patterns (ART; Hsu and Chien, 2007). One of the standouts is Leabra, which recognized a hundred object categories rendered in different poses and under various 2D transformations (O'Reilly et al., 2013).

Some vision applications require not only spatial but also temporal processing. The most common scenario is detection, identification, and tracking of moving objects. For instance, IMA can recognize up to five different persons (Rogers and Wilkes, 2000) and ADAPT tracks vehicles in videos (Benjamin et al., 2015). CARACaS further analyzes the tracked movements to recognize the actions performed by the human-like agents in a 3D simulation (Huntsberger, 2011a). ART applies spatio-temporal analysis to detect the changes in satellite imagery (Carpenter et al., 2001).

In addition to visual data, some cognitive architectures can process other types of signals. ART is one of the architectures frequently used for this purpose in place of other machine learning approaches, such as PCA and neural networks. ART has performed successfully on over a hundred of different tasks, including anomaly detection for a range of industrial applications: sensor malfunction (Aradhye et al., 2004), energy system faults (Ferreira et al., 2006), milling tool failure (Tansel et al., 1995), fish freshness analysis (Gil et al., 2008), sensor fusion for robotics (Martens et al., 1998b), and speech processing from audio (Goldinger and Azuma, 2003).

9.3.2 Playing games and solving puzzles

Board games and puzzles have been a staple of research in both AI and cognitive architectures. Some of the earliest AI applications focused on playing

checkers (Samuel, 1967), as both a test bed for developing and optimizing new algorithms and a way to attract public attention. After years of progress, computer performance has now surpassed that of most capable human players (Silver et al., 2018).

Although not as competent as their specialized AI counterparts, cognitive architectures too can play games and solve puzzles, all while capturing aspects of human perception and decision-making processes. For example, CHREST matched the human ability to detect when a king is attacked in chess (referred to as a check position) (Smith et al., 2009), Sigma made decisions on a human timescale when playing Othello (Rosenbloom et al., 2015a), ACT-R learned to play backgammon on the level of common machine learning approaches but with much less training data (Sanner et al., 2000), FORR learned to play 18 common board games against imperfect opponents (Epstein, 1994), STAR ranked 18th in the world among human players of a popular online video game, Canabalt (Kotseruba and Tsotsos, 2017), and Soar demonstrated knowledge reuse when learning new two-player games and solving puzzles (Kirk and Laird, 2016).

9.3.3 General problem-solving

Besides games, cognitive architectures have been applied to general problem-solving. Many of these tasks are similar to the abstract reasoning tasks described in the previous section, but are more realistic. Examples include select Advanced Placement Physics problems (Companion; Klenk and Forbus, 2007), patient diagnosis (Leabra, Pauli and O'Reilly, 2008; OSCAR, Pollock and Hosea, 1995), dynamic stocks and flows (ACT-R; Halbrügge, 2010), negotiation problems (SHRUTI, Wendelken and Shastri, 2002; Sigma, Pynadath et al., 2014; LISA, Großmann et al., 2012; DUAL, Petkov et al., 2011), and planning problems in various domains (CIRCA, Ha and Musliner, 2002; PRODIGY, Wang and Carbonell, 1994; OMAR, Deutsch et al., 1998).

9.3.4 Language understanding

Understanding the meaning of spoken or written commands is important for many practical applications. In particular, much work has been dedicated to dealing with the inherent ambiguity of natural language. Some examples include anaphora resolution (ACT-R; Pyke et al., 2007) and analogical inference (DUAL, Kokinov, 1994a; LISA, Krawczyk et al., 2004), text comprehension (Companion; Ribeiro et al., 2019), and filling in implicit information via causal reasoning (Meta-AQUA; Cox and Ram, 1994).

9.4 Procedural tasks

9.4.1 Navigation

The ability to move from one place to another is demonstrated by many architectures in simulated and real environments and with aerial and ground robots. Simulated environments range from simple 2D grid-like worlds to 3D simulations of rooms or corridors,which are empty or with few obstacles. The actions performed are usually a combination of navigating to the goal,

wandering, wall-following, and obstacle avoidance. For example, FORR was tested in a small simulated maze (Epstein, 1997), the Clarion agent navigated a 2D minefield (Sun et al., 1998), and Soar moved inside a Pacman-like simulation (Marinier III et al., 2009) as well as 3D indoor space with multiple rooms (Könik and Laird, 2006). Physical indoor environments are represented by research labs and university building hallways (Subsumption, Brooks, 1986; SASE, Zhang et al., 2005; DIARC, Talamadupula et al., 2010). Outdoor environments are less common. Among notable examples are RCS controlling an unmanned vehicle driving off-road (Albus, 2002) and robots for interplanetary exploration tested on rough terrains, such as Ratler (Simmons, 1995) and Ambler (Krotkov and Simmons, 1996), both controlled by TCA.

Other environments and robots have also been considered, such as performing basic maneuvers using a simulated vehicle (ICARUS, Choi, 2009; ACT-R, Haring et al., 2012; IMPRINT, Kandemir et al., 2018), simulated aircraft (Soar, Gunetti et al., 2010; ICARUS, Langley, 1996a; ACT-R, Gunzelmann and Gluck, 2009; CIRCA, Atkins and Durfee, 1997), and an underwater vehicle, simulated (Clarion, Sun and Peterson, 1998b; ART, Tan et al., 2008) or real (CARACaS; Austin and Stokey, 2011).

9.4.2 Reaching and object manipulation

Most abilities in this area are fairly simple, such as approaching a salient object located nearby (MDB; Bellas et al., 2010b), foraging (DAC, Verschure et al., 2003; BBD, Krichmar, 2000; ACT-R, Harrison et al., 2003), and following arbitrary moving objects (Subsumption; Horswill and Brooks, 1988).

Object manipulation is more complex as it requires precision and dexterity. Some implementations are in simulation (e.g. picking objects from a conveyor belt by the CIRCA-controlled robotic arm (Musliner et al., 1994), stacking blocks demonstrated by SAL (Herd et al., 2014), or pointing at objects by BECCA (Malone et al., 2014) and some are performed using physical robots. For example, the MDB architecture enabled a Baxter robot to pick up objects from a platform (Salgado et al., 2016) and DIARC was used to implement grasping behavior on a PR2 robot (Wilson et al., 2016).

A popular demo task combining navigation and object manipulation abilities is a fetch-and-carry task. One of the first demonstrations of this kind was a Subsumption robot called Herbert that wandered into the offices and used laser sensors to detect and steal empty soda cans from the desks (although not very reliably) (Brooks, 1989). Other similar demos followed shortly. CHIP based on the AAA architecture visually identified and located crumpled paper and empty soda cans on the office floor, picked them up with a gripper and took them to the trash can (Firby et al., 1995). A TCA-controlled robot collected empty cups and also periodically recharged its battery (Simmons, 1989). A variation on this task is the collection and delivery of rock samples by the Rocky III autonomous rover (ATLANTIS; Gat, 1991a).

9.4.3 Safety-critical tasks

The activities in this category are often challenging and take place under significant time pressure in rapidly changing circumstances, and thus place high demands on human perception, memory, and decision-making. These

activities include air traffic control (MIDAS, Corker, 1999; APEX, Freed and Remington, 1997; ACT-R, Lebiere, 2006), driving (ACT-R; Salvucci, 2002), military operations (COGNET; Ryder and Zachary, 1991), space missions (PRS; Ingrand and Georgeff, 1990), etc.

A combination of perceptual and decision-making abilities and practical necessity of safety-critical tasks made them an attractive target for modeling in a number of cognitive architectures. While the outputs of the models are similar—a sequence of decisions and actions that a human operator would undertake in a given situation—they serve different purposes. Cognitive architectures for human performance modeling, such as APEX, MIDAS, OMAR (and its distributed version D-OMAR), MHP, and COGNET, are primarily concerned with the human-computer interaction aspect of the task, seeking to improve the interfaces and, consequently, reduce human error. On the other hand, the psychologically inspired architectures ACT-R, EPIC, and Soar use the ATC task as a challenging real-world activity against which to validate hypotheses (Wray and Chong, 2003; Crossman et al., 2004b).

9.5 Interactive and social tasks

Most social tasks involve interaction, but not all interactions are social. Even interpreted broadly, common definitions of "social interaction" require capabilities, such as consciousness, self-identity, intentionality, other-directedness, and communication (Cerulo, 2009). Even though none of the existing cognitive architectures possess all required qualities, many of them can operate in an environment populated by others, communicate with them, and respond in ways that are not fully preprogrammed or reactive.

9.5.1 Virtual assistants

Cognitive architectures have been used to create software agents for assisting people in their personal and professional activities. The BB1-based Patient Advocate helped patients by monitoring and visualizing personalized health data (e.g. exercise, blood glucose) and suggesting changes in nutrition or daily routine according to the prescribed treatments (Miksch et al., 1997). VMattie (precursor of LIDA) was designed to reduce the workload of departmental administrators. It could maintain mailing lists and schedule and announce upcoming seminars after receiving emails from members of the department (Stan and Franklin, 2000). Disciple provided, perhaps, the most personalized assistance since not only could it answer questions and provide suggestions, but it also was able to learn from interactions with human users. Disciple agents have been used in a variety of domains, such as creating engineering designs (Dybala et al., 1996), planning and critiquing plans for combat operations (Boicu et al., 2000), or choosing a PhD supervisor (Boicu and Tecuci, 2004). Needless to say, these assistants were prototypes with many limitations. They could not process speech or complex written instructions, preferring input via graphical user interfaces or using formalized languages instead. In addition, assistants lacked proactiveness and autonomy, often relying on user feedback to continue operation. As a result, interaction with these systems could be slow and cumbersome.

9.5.2 Robot assistants

Robot assistants are meant to perform either the tasks that are routine and avoided by humans or the tasks that the human users cannot do themselves (e.g. due to disability or not being physically present). The second category is represented by ISAC, a robot based on the IMA architecture, designed to assist a disabled person during meals (Bagchi and Kawamura, 1992) and a Robotic Construction Crew prototype for building an outpost on the Moon prior to human astronaut arrival. The latter is a set of two robots controlled by CARACaS to pick up, carry, and stack metal beams (Stroupe et al., 2006). Otherwise, delivery or fetching objects is the most common task for service robots. One of the early attempts of this kind is Xavier, an office delivery robot prototype controlled by the TCA architecture. The first demo did not include the actual delivery tasks: the robot took commands via a simple web-based interface to go to different offices to say "hello," take a picture, or tell a knock-knock joke (Simmons et al., 2002). Its next task was more practical: go to the coffee shop located in the building, wait in line, and place an order. Once the coffee was placed in the robot's cupholder, it went back to the user (Nakauchi and Simmons, 1999). Although these tasks involved some interaction, it was rather limited and much of the behavior was preprogrammed. A more realistic scenario was used to test CORTEX, which powered a salesman robot (Bandera et al., 2017). Its task was to attract people at the exhibition hall to attend an advertising stand. The robot approached the people in the hall, introduced itself, and led them to the stand if they showed interest. The robot's ability to have a conversation and read social cues was rather limited, and a touch screen was added later to improve communication.

9.5.3 Social robots

Social robots do not run errands but provide entertainment or comfort. The lack of practical utility is often compensated by a more advanced social ability—following norms, trying to understand others, and expressing its own desires. Perhaps, the best known social robot is Kismet, embodied as an expressive robotic head. It could perceive some human social cues by detecting motion and tone of speech directed at it, as well as conveying its own intentions mainly through facial movements (Breazeal, 1998a). Kismet's older sibling Cog, developed in the same MIT lab, has a humanoid torso with a head. Having a more expressive body, Cog could follow gaze, recognize declarative pointing, and imitate head nodding (Scassellati, 1998).

The software social agents focus more on seamless verbal interaction. A Cybercafe waiter (based on the BB1 architecture) recognized customers who were ready to order, knew to wait for a response, and improvised during conversation (Rousseau and Moulin, 1997). Another example, a Ymir-based agent, Askur, explored taking turns during dialog. It used prosody patterns to recognize when the other person finished speaking and adjusted to their individual preferences for pauses between utterances (Jonsdottir and Thórisson, 2013).

9.6 Real-world and commercial applications

Applications presented in the previous sections are quite limited, and lack the robustness required for real-world use cases. Here, we will look at some past successful uses of cognitive architectures in industry and for developing commercial products.

Autonomous driving. The 4D/RCS architecture was developed as part of the Department of Defense (DoD) Unmanned Ground Vehicle (UGV) program, a multiyear effort to advance the technology for autonomous mobile robots that could traverse rugged terrain and cooperate with other manned and unmanned vehicles. For this purpose, RCS integrated the 4-D approach to machine vision proposed by the German pioneer of driverless cars, Ernst Dickmanns (Dickmanns, 1990). The platform provided for test was a custom designed eXperimental Unmanned Vehicle (XUV)—a heavy robotic scout with four-wheel drive and four-wheel steering, weighing up to 2,800 lb (1,300 kg), and equipped with LIDAR in addition to color and infrared stereo cameras for day and night driving, respectively. Demo III involved an extensive series of field experiments. The vehicles had to traverse deserts, mountain foothills, and urban streets (represented by military housing areas). Overall, XUV traveled over 550 km through varying terrains, in day and night time, and under clear weather, rain, and snow. The vehicle operated autonomously for 90% of the time and distance traveled. A human operator intervened remotely when the vehicle got stuck, e.g. due to terrain conditions, such as sand, a steep slope, or dense vegetation, or when it failed to reach the next waypoint after multiple attempts (Shoemaker and Bornstein, 1998).

Interface design and process analysis. MIDAS is a human performance model developed jointly by the US Army and NASA since 1983. It provides a set of tools for rapid prototyping of workstations and modeling how it will be operated by humans. MIDAS, currently in its fifth version, has been used for multiple military and civil purposes: the analysis of military missions performed by helicopters and autonomous underwater vehicles, the design of the 911 dispatch console and nuclear power plant consoles, testing new designs of military clothing and equipment during helicopter missions, the analysis of air traffic control procedures, and the redesign of shuttle cockpit (Hart et al., 2001). More recently, MIDAS was used to evaluate the Next Generation Air Transportation System (NextGen), a plan to modernize the National Airspace System (NAS) developed by NASA together with the Federal Aviation Administration (FAA). A high-fidelity model of a two-person commercial aircraft crew was constructed based on the task analysis and walkthroughs provided by pilots and air traffic controllers. This model was validated on the current procedures for approach and landing and then applied to off-nominal events (e.g. high winds, rogue aircraft on the runway) to test the effect of the NextGen concepts on pilot performance (Gore et al., 2011).

Space mission support. The 3T architecture was developed at NASA to automate and assist astronauts in various tasks, such as operating the shuttle manipulator (Dorais et al., 1999) and the life support system (Bonasso et al., 2003). The 3T-controlled life support system initially was designed to manage concentrations of CO_2 and O_2 in the greenhouse chamber and crew habitat atmospheres during space missions (Schreckenghost et al., 1998) and later applied to the water recovery system intended for use at International

Space Station (Bonasso et al., 2003). Several tests were conducted to validate operation of the life support system: first, a 15-day test with one person in 1995, then, a 91-day test with a 4-person crew in 1997 (Schreckenghost et al., 1998), and a 450-day test for the water recovery system in 1999 (Bonasso et al., 2003), followed by more tests in 2000–2002. Throughout, 3T operated autonomously with minimal manual interventions.

Combat flight simulation. TacAir-Soar is a system that emulates the behavior of human pilots during the missions on a fixed-wing aircraft. It is intended for use in simulated military training exercises in place of human-controlled agents. After five years in development and numerous demonstrations, TacAir-Soar was tested in Synthetic Theater of War (STOW) in 1997 and a smaller training exercise in the following year. STOW demonstrated the utility of Soar in a large-scale battlefield simulation that spanned 500×775 square kilometers with 13,500 buildings and over 20,000 objects (roads, bridges, vegetation, etc.). Over 3,700 agents represented opposing forces, with Soar controlling 100 agents simultaneously in real time, each taking in 200 data points as input and producing over 30 different types of outputs. Overall, the system completed over 700 flight missions, 90 minutes to 8 hours in duration. Nearly 99% of missions were successfully launched and 95% of them completed (Jones et al., 1999).

Assignment scheduling system. IDA (a precursor of LIDA) was developed to automate the duties of human detailers in the US Navy who assign jobs to the sailors (McCauley and Franklin, 2002). To this end, IDA communicated with the sailors by email in unconstrained English to find their preferences and match them with available jobs, all while following numerous organizational policies. Often, after receiving an email with a job offer, the sailors declined the assignment and requested a different one or specified additional requirements. These negotiations often took several emails and IDA's job was to correctly evaluate the context, decide on the appropriate response, and convey it to the sailor (Franklin, 2003). The in-house testing by the Navy determined that the system performed on par with the human detailers and as of 2009 the system was still in operation (Franklin et al., 2009).

Engineering design retrieval system. In the 1990s, Boeing needed a system for engineering design retrieval. At the time, it operated with a database of over 55,000 2D and 95,000 3D models of various parts which were searched using a classification and coding system that was time-consuming to maintain and difficult to update with new parts. An in-house research team used self-organizing ART networks to build a neural information retrieval system (NIRS) by training it on the database. During a search, the users could also vary the level of similarity controlled by the ART vigilance parameter (Escobedo et al., 1993). The system was made available to thousands of Boeing engineers in the state of Washington, with plans to extend it to the entire company (Smith et al., 1997), however, there is no information on whether it was done.

It is not difficult to notice that the most advanced applications of cognitive architectures are for military or government needs, and the most recent ones are over two decades old. Commercial products based on cognitive architectures are both more recent and aimed at the general public. However, with proprietary software, it is often difficult to establish how much the end product

actually relies on the principles and implementation of the corresponding cognitive architecture. Nevertheless, we list some examples in chronological order.

Cognitive Tutor. The commercial version of the Cognitive Tutor grew out of nearly twenty years of research on ACT-R at Carnegie Mellon University (Ritter et al., 2007). In 1998, after several successful pilots in high-schools, the tutor became commercially available through Carnegie Learning Inc. Cognitive tutors were available for algebra and geometry courses and allowed students to ask questions and track progress. By 2011, the company served almost 3,000 schools (Kelkar, 2022). These days, after a series of acquisitions and leadership changes, the focus of Carnegie Learning appears to have changed. In 2023, the company announced a new math tutor LiveHint AI, based on the large language model trained on the proprietary data.

Soar Technology, LLC. Soar Technology or SoarTech is a spin-off of the Soar cognitive architecture developed at the University of Michigan. SoarTech was started in 1998 to further develop TacAir-Soar mentioned earlier and has since been supplying a number of military clients. Commercial products include realistic computer-generated scenarios and agents for training, virtual assistants, and tools for controlling swarms of unmanned systems.[1]

Open Sesame! In 1993, a division of Charles River Analytics launched one of the first virtual assistants designed for the Macintosh platform (Fink and Kobsa, 2000). Its goal was to eliminate mundane and routine interactions with the computer. It monitored user actions, detected patterns, and automated them. Open Sesame! detected time- and event-based triggers, e.g. opening an email or news app every morning or opening a folder with notes before writing a text document. This ability was enabled by the ART networks that categorized high level interface events in a self-supervised fashion and adapted to the user preferences (Caglayan et al., 1997). Open Sesame! saw an initial success, shipping to over 35,000 clients, however, its development halted in 1996 due to issues of a technical (insufficient operating system support) and non-technical (lack of situatedness) nature (Hoyle and Lueg, 1997).

Roomba. By far, the most commercially successful product that resulted from research in cognitive architecture is a cleaning robot called Roomba. The first prototype called Rug Warrior ran on the Subsumption architecture developed at MIT (Jones et al., 1998, p. 293). Its successors went on to sell millions of units after commercial release in 2002.

Jibo. Another product, also from MIT, is the social robot Jibo, a descendant of Kismet. Jibo had an expressive head and an articulated body and was meant to be a companion robot for the whole family. In the demos publicized by the company, the robot could recognize members of the household, coordinate their calendars, answer questions, and act as a photographer for family celebrations. Jibo went on sale in 2017, however, not many units were sold due to its high price, limited capabilities, and issues with hardware. The company eventually ceased operations and hardware support in 2020 due to intense competition from voice-based assistants that lacked physical presence but were cheaper and more useful for everyday tasks.

[1] https://soartech.com/capabilities/

Lastly (and curiously), there is a cognitive architecture that grew out of a commercial product. Novamente LLC was a company founded in 2001 that designed and developed virtual agents with social abilities for virtual worlds, computer games, and training simulators. However, in 2008 it donated the code base for its Novamente Cognitive Engine (NCE) to CogPrime, which is still actively developed.

9.7 Summary

- Collectively, cognitive architectures show an extensive breadth and depth of research and practical applications. We identified over 1,000 applications and grouped them into five categories: abstract, perception & reasoning, procedural, interactive & social, and real-world & commercial.
- Abstract tasks are simplified and the abstracted versions of the human activities designed to isolate and study specific cognitive abilities in experimental settings. Some of the most common examples of such tasks are visual search, recalling lists of items, problem-solving, and performing two or more of these tasks concurrently.
- Perception/reasoning tasks involve perceptual processing and inference on realistic data, for example, pattern recognition, object classification, playing games, solving puzzles, and understanding natural language.
- Procedural tasks build on reasoning and perception to enable physical actions, such as navigation and object manipulation, in simulated or real environments.
- Interactive and social tasks are the most advanced as they utilize perception, reasoning, and decision-making to enable assistive functions, virtually or via physical robots.
- Several cognitive architectures have been tested on real-world tasks, such as autonomous driving, space mission control, and decision-support systems, and some even led to commercial products, like cognitive tutor based on ACT-R and robot vacuum Roomba that grew out of Subsumption.
- The task categories listed above are not distributed evenly. Aside from several exceptions, most architectures are focused on narrow application domains. Overall, abstract, procedural, and basic perception and reasoning tasks comprise the majority of applications of cognitive architectures. More advanced applications that combine these abilities are not as common. Even less so are real-world and commercial applications.

10 Evaluating Cognitive Architectures

In the previous chapter, we looked at what cognitive architectures can do. Here, we will focus on how to evaluate their performance and what methods can be used for this purpose. In cognitive science and computer science, qualitative and quantitative analysis across a variety of tasks and domains have been instrumental for determining and comparing the abilities of different methods. However, evaluating cognitive architectures is challenging due to their complexity. Recalling the discussion in Chapter 1, cognitive architectures bridge several levels of abstraction. Therefore, most architectures contain elements of verbal theory, computational and algorithmic specifications, and implementation in a specific programming language or on a specific piece of hardware. Thus, a single evaluation method will not be sufficient. Furthermore, various elements of the architectures are not always fully separable or fully specified, complicating analysis and evaluation.

Despite the challenges, there are ways of evaluating complex intelligent systems. In this chapter, following Hernández-Orallo (2017), we will consider two broad groups of evaluation methods—task-based and ability-based. The task-based methods include a large arsenal of tools developed in artificial intelligence (AI) and cognitive science. Such methods are better suited for measuring performance of individual instances and models on specific tasks. A second option is evaluating the system's skills and intelligence by considering them as a whole. In this chapter, we discuss both groups of approaches:

Section 10.1 overviews task-based evaluation methods. Because there are many such methods, they are further subdivided into groups, such as qualitative vs. quantitative, non-comparative vs. comparative, etc.

Section 10.2 describes proposals for ability-based methods that assess whether a given intelligent system has certain cognitive abilities. These include behavioral tests of intelligence (e.g. Turing test and its variants, psychometrics) and approaches for evaluating cognitive plausibility.

10.1 Task-based evaluation

Task-based evaluation, as the name implies, relies on specific tasks with known outcomes. There are many ways of dividing task-based evaluation methods across multiple dimensions, such as qualitative or quantitative, analytical or empirical, individual or comparative, benchmark-driven or example-based, and automated or administered by humans. Each type of method has its uses, advantages, and limitations, which will be discussed below.

First, we will look at the diagram in Figure 10.1A that summarizes all task-based evaluation methods used to assess the performance of cognitive architectures on different applications discussed in the previous chapter. Overall, qualitative evaluation is quite common and is

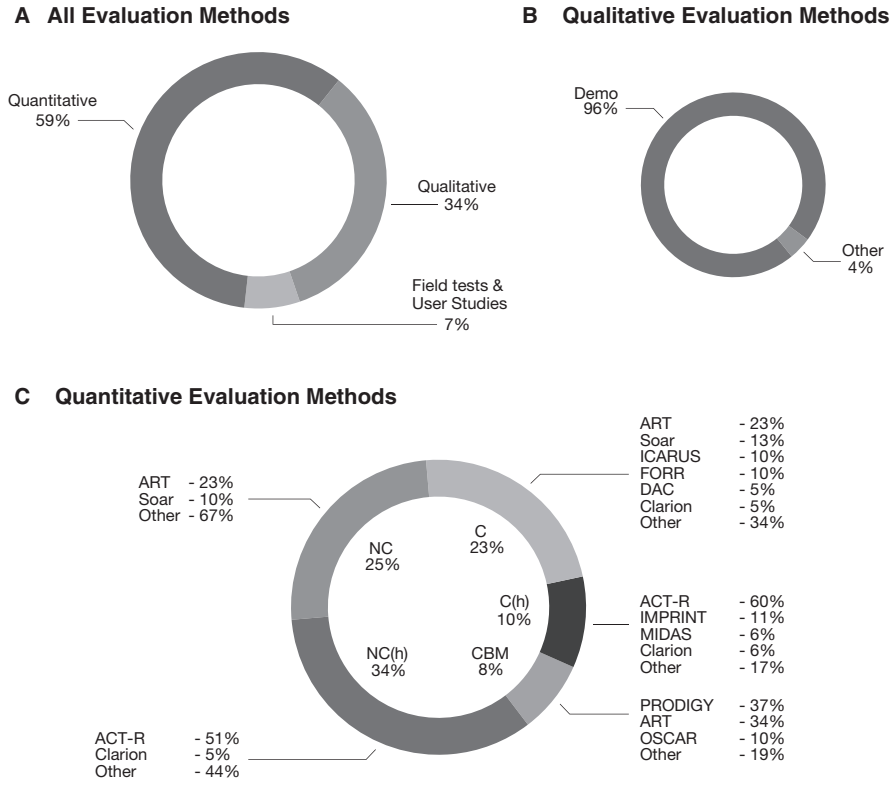

Fig. 10.1 An overview of evaluation methods. A) A distribution of quantitative and qualitative methods, as well as field tests and user studies. B) A distribution of qualitative methods. C) A distribution of quantitative evaluation methods. The following abbreviations are used: NC—non-comparative, i.e. the method is evaluated individually; NC(h)—non-comparative evaluation on human data; C—comparative evaluation; C(h)—comparative evaluation against human data; CBM—comparative benchmark evaluation, i.e. unlike all previous case-based methods, a benchmark is used. For each category, the percentages are normalized and cognitive architectures with the share above 5% of the total in the category are shown. All statistics are computed based on the set of 800 implementations of tasks performed by cognitive architectures and for which evaluation has been conducted.

demonstrations, i.e. narrative descriptions, optionally accompanied by illustrations. Field tests, and user studies are the smallest group, in line with the rarity of the real-world applications of cognitive architectures (see Chapter 9). Although quantitative assessment comprises more than half of all evaluations conducted, it is mostly non-comparative, i.e. the cognitive architectures are evaluated individually, often on small sets of examples or experimental data from small groups of human subjects (see Figure 10.1C).

The type of qualitative and quantitative evaluation largely depends on the type of architecture. For example, psychologically motivated architectures are more often compared to the results of human psychological experiments, whereas direct comparisons against human data are rare for robotic architectures. Instead, they are assessed through human user studies and field tests. The diversity of evaluation methods used for assessing individual cognitive architectures is correlated with the diversity of tasks that they can perform,

because evaluation methods are often specific to the task and domain. As a result, about half of all cognitive architectures use a combination of three or more qualitative and quantitative methods.

10.1.1 Non-comparative evaluation

Non-comparative evaluation methods are a family of evaluation approaches that apply to individual architectures and tasks. As the name suggests, comparisons to other similar systems are not made. Instead, the focus is on investigating characteristics of the system, its computational cost, and effectiveness at performing the given task.

Demonstration

Demonstrations are extensive narrative descriptions of the task execution, often accompanied by details of internal processes, diagrams, visualizations, or code listings. If available, photos or links to video recordings of the system in action serve as additional illustrations. The primary goal of narrative demonstrations is to provide an in-depth explanation of the inner workings of the architecture and proof of its feasibility for a particular task. Other aspects of performance, such as complexity, effectiveness, efficiency, and limitations of the approach, are either discussed in qualitative terms or omitted altogether.

Although demos are helpful for understanding the inner workings of the architecture, they have significant limitations due to their qualitative nature. First, the descriptions are usually given from the authors' subjective point of view. As such, they comprise the best of a system's performance and may not fully reflect the true capabilities and constraints of the approach. Second, while tasks used for demo may be complex, their number is usually small. Thus, the potential for generalization is difficult to assess. Finally, narrow selection of tasks and lack of quantitative data prevent comparisons on other tasks and against other approaches. Given these issues, ideally, demos should be combined with the quantitative evaluation methods but often are not; 70% of the applications that feature demos do not use any other evaluation.

Fit to human data

One of the main goals of cognitive architecture research is to replicate and explain what happens inside the human mind, therefore comparison to human performance is a natural choice for evaluation. First, a task is selected for which human data exists, or an experiment is conducted with human participants to gather data. Then an instance of the cognitive architecture is constructed for the task, with inputs and outputs approximating stimuli presented to the human subjects and their responses, respectively. Finally, the output of the model is compared to the human data. A good fit supports the claim that the system behaved in a human-like manner when performing the task. The examination of the intermediate representations can further confirm that the architecture works in line with known theories or reveal previously unknown phenomena for further research and experimentation.

For the psychologically motivated cognitive architectures, such as ACT-R, CHREST, DUAL, and EPIC, comparing the predictions of the models to the results of human experiments is a prominent part of their evaluation strategy. ACT-R stands out in this group both in terms of the absolute number of

human experiments replicated (over 130) and in relative terms (two-thirds of all ACT-R evaluations are based on psychological experiments, compared to less than a quarter for other architectures). Overall, since the 1970s, hundreds of experiments were reproduced in dozens of cognitive architectures. Although modeling human data is well established in cognitive science, such evaluations must be judged critically due to the limitations that can diminish the meaning and significance of the results. The issues can be grouped as related to data, models, and the fit between the two.

Sources of human data. Research on cognitive architectures relies on data from numerous psychological studies. About half of these results come from published experiments, and the other half are the results from studies designed and conducted to be replicated in cognitive architectures. There are trade-offs associated with each. Using data from the existing publications saves labor, but can create additional difficulties. For example, it may be difficult to replicate an experimental setup if its description is incomplete. In addition, if raw experimental data is not available, statistical analysis may be limited (more on this later). While designing the experiment and gathering data for the purpose of validating the architecture can alleviate some of these problems, it may inadvertently introduce bias if the task and experimental conditions are more aligned with what the cognitive architecture can do.

Validity of data. Independent of the experiment design, there is also a chance that the discovered effect may not exist (a false positive) or exist only under certain conditions, in which case the model and the evaluation are meaningless at best and misleading at worst. This is not a new concern. In fact, the issues with the validity of findings in psychology has been debated for many decades (Bakan, 1966; Rossi, 1990).

The "validity crisis" in psychology and cognitive science has many contributing factors, such as lack of replicability studies (Francis, 2012), misunderstanding of replicability (Schmidt and Oh, 2016), low statistical power of the results (Morey and Lakens, 2017), widespread p-hacking[1] (Simmons et al., 2016), bias toward publishing only positive findings (Ioannidis, 2012), and even intentional fabrication of results (Fanelli, 2012).

Because the causes are numerous, there is no universal remedy. Over the years, long lists of best practices have been formulated, focusing on improving collection, sharing, and statistical analysis of data (Lishner, 2015; Lee et al., 2019). Meta-studies, data and model availability, preregistering the experiments, and other countermeasures are becoming more common in psychology and cognitive science. Although validity concerns apply to research on cognitive architectures as well, the problem is not widely acknowledged and best practices are not applied consistently in this field.

Model fidelity. Cognitive architectures do not model all aspects of human cognition with equal fidelity. Perception and motor control, in particular, are often simplified the most. Theoretically, fidelity will suffer more when using

[1] P-hacking refers to various misuses of the p-value, which is most commonly used in the null hypothesis significance testing as a statistic that measures strength of evidence against null hypothesis (Schervish, 1996). Since p-value is one of the main proofs for the validity of findings, it is susceptible to manipulation. P-hacking is widespread and takes many forms, from selective reporting of only significant results to various postanalysis techniques to boost p-values, and stopping analysis once desired significance is achieved (Head et al., 2015).

the results of the previously published experiments, since the description of the methods and instruments may not be sufficient for the full reconstruction. However, simplifications happen even when using data collected for the purpose of modeling. For example, an experiment for validating a DUAL model of episodic memory recall required human subjects to view images but the model itself lacked perception (Kokinov et al., 2007). Similarly, reconstruction of the recorded data regarding bonus negotiations in Clarion focused only on affect, omitting the conversational component (Allen and Sun, 2016). To evaluate the model, self-reported values of affect were mapped to Clarion's drives.

When the authors have control over the experiment design, they can also modify it for a better fit to the abilities of the cognitive architecture. For modeling the Tower of Hanoi in ACT-R, Anderson and Douglass (2001) made the following changes: to simplify perception and motor response, the puzzle was presented as a 2D schematic on the computer screen and mouse clicks were used to move disks. To expose problem-solving strategies, human participants were asked not only to move the disks to destination pegs, but also to specify goals that they could not act on yet.

Finally, there is the issue of adapting the output of the model to better match the results of human experiments. One concern is timing of the events, particularly when response times are being compared. Since cognitive architectures run on computer hardware, some processes (e.g. visual processing and motor responses) could take longer than those of humans, while others (e.g. symbol manipulations) may run significantly faster. One solution to this issue is to set constant timings for the events based on the estimated values from human data, as has been done in ACT-R, EPIC, CHREST, MHP, and few others. Another is to postprocess the output of the cognitive architecture. For instance, CPU times of the Soar model were averaged over multiple trials and scaled to better match human response times (Lathrop and Laird, 2007). In ACT-R model, the retrieval time parameter was increased to compensate for slow execution time of the motor commands (Harrison and Trafton, 2010).

In sum, for a variety of reasons, replicating the exact conditions, stimuli, and output of human experiments is almost never feasible and approximations are inevitable. However, their extent is in many cases unknown because both experimental setup and its simulated version are often not provided in sufficient detail. The effects of modeling simplifications on the results are even more difficult to estimate, and we are not aware of any studies that attempted to do so.

Fit to human data. The correspondence between the human data and model output can be shown in different ways. Sun and Ling (1998) list, in the order of increasing precision, behavioral, qualitative, and quantitative fit. Behavioral match is the easiest to satisfy—all that is needed is a rough correspondence between the model outputs and human data under similar conditions. Qualitative and quantitative fit require relative and absolute fit to the data in numerical terms, respectively, which is more difficult to attain. Most of the models we considered achieve at least qualitative correspondence, i.e. the same trends in performance and errors as human subjects did under various conditions. Fewer can show a quantitative match in absolute numerical terms. At any rate, given the issues with model fidelity discussed earlier, such matches would be difficult to ascertain.

Presuming it can be done, is showing a good fit to data enough? The answer can be found in the psychology literature, where goodness of fit continues to be a dominant but also controversial evaluation approach. The fact that more complex psychological models present a new set of challenges for evaluation has been recognized for at least sixty years (Grant, 1962) and later summarized in an influential work by Roberts and Pashler (2000) as the following three issues:

- *Model flexibility.* If a model has adjustable parameters, then showing a good fit to the data for a single set of parameters is insufficient. Instead, the parameters should be varied over their range to determine what other relations between the measures can and cannot be modeled.
- *Data variability.* Without knowing the range of plausible values of observations, good fit is meaningless. One should determine the variability of the data along the dimensions that are constrained by the model. Only if they are consistent, should the model fit be considered good.
- *Fit likelihood.* If a theory can fit any plausible result, the theory is not falsifiable and fit to the data is not as impressive. Thus, there should be plausible outcomes not predicted by the model.

These concerns are valid not only for psychological models but for cognitive architectures as well, as acknowledged by many in the field (Simon, 1992; Ritter et al., 2003; Sun et al., 2005; Gluck and Pew, 2006; Halbrügge, 2007). For example, Halbrügge (2007) notes that cognitive architectures present an additional set of challenges, namely:

- Possible mismatch between the theory and its implementation in the cognitive architecture;
- Limited interpretability of the models, since software is often unavailable or difficult to understand;
- Introduction of additional assumptions, which may contribute to the performance or fit to data more than the underlying theory;
- Difficulty of establishing falsibiality. As complete Turing machines, cognitive architectures can perform any computation; if a cognitive architecture can model an empirical effect, it can in principle model its negation as well. Thus, only specific instances or assumptions can be falsified, not the entire architecture.

Only some of these concerns have been addressed in the literature on cognitive architectures, specifically determining the complexity of the models, generalization to other tasks or datasets, falsifiability, and implementation.

- *Model complexity.* Complexity of the models is usually discussed in terms of free parameters. A model with only a few or zero free parameters is seen as a more convincing proof of the validity of the cognitive architecture because the fit to data can be attributed to the theory and assumptions of the model rather than parameter manipulation.

  What constitutes a free parameter in a cognitive architecture is not very clear. The consensus appears to be that free parameters are those that are a) set on the architectural level and used across many instances (e.g. global settings, such as learning rate, memory decay, duration of cognitive cycle) and b) are varied systematically or to match experimental conditions. For production systems, the free parameter set could also include production rules (Simon, 1992) as well as the memory elements (chunks) used

to represent the problem and activated during problem-solving (Baker et al., 2003). In addition, the algorithmic structure of the model introduces additional degrees of freedom that are difficult to account for, as shown in the experiment by Halbrügge (2007).

In practice, select few architectures perform any kind of complexity analysis and in doing so usually consider only global parameters, ignoring other sources of complexity, such as memory contents or trainable weights of connectionist components, if present.

- *Model generalization.* Generalization is the ability of the model to respond adequately to unseen data from the same task or perform a different task. The fact that many cognitive architectures have shown a good fit to multiple tasks and datasets using the models with a few parameters is often used as a proof of their generalizability. What these statements often omit is whether the same version of the architecture and the same implementation of the model was used. Particularly, for the projects that span decades, changes accumulated over time may be significant. Even optimizations to the code, not related to the core architectural mechanisms, may change the behavior of the model.
- *Model falsifiability.* Falsifiability of architectures or their instances remains largely unaddressed. Only several case studies have been conducted, and most do not refute the issues mentioned earlier. Specifically, Schultheis (2009) showed that ACT-R put too few constraints on computation and could model any function. In addition, the existence of numerous software variants of ACT-R and further modifications and assumptions made in individual models made it nearly impossible to ascertain the compliance of models to theory. Similar issues have been pointed out for Soar (Hunt and Luce, 1992; Cooper, 2006) and likely apply to other projects.
- *Implementation.* Lane and Gobet (2012), using CHREST as an example, argue for adopting software engineering practices to address the problems of theory and implementation mismatch, versioning, and reproducibility in cognitive architectures. Specifically, the benefits of test-driven development and refactoring are demonstrated. For example, carefully designed tests help identify failures in implementation, ensure consistency across different versions of software, and identify gaps in theory. Refactoring, in turn, has little effect on the functionality but helps maintain quality of design as the software base evolves.

Effectiveness evaluation

Effectiveness evaluation quantifies how well a cognitive architecture performs a particular task with respect to the numeric metrics and measures that have specific interpretation. For example, accuracy and error rates are used for classification and recognition tasks, task completion rates are used for various robotic tasks, and positioning errors for object manipulation tasks. Besides metrics, quantitative evaluation requires a set of test cases with known outcomes. The test set must be of appropriate size: a set too small will likely be not representative, but a large set may be difficult to collect and later difficult to examine. Selection of test cases also matters: they may be sampled randomly from a larger population, selected systematically, or chosen to represent a typical case. Lastly, it is important that the operational domain is precisely defined.

The majority of cognitive architectures are tested on a few dozen test cases—images, data samples, or simulation runs, depending on the application. Selection process is often not discussed explicitly, however can be inferred from the literature to some extent. Systematic exploration of a range of values is the most common, followed by a selection of typical cases, as deemed by the authors. In rare cases, testing was limited to several ad hoc examples. The only exception is the Leabra model for object recognition tested on a synthetic benchmark CU3D-100 comprised of 100 object categories (with 9–10 examples of each) and controlled variability in pose and illumination (O'Reilly et al., 2013).

An obvious characteristic of individual evaluation is that it does not require comparisons to other models. It is somewhat justified when certain expectations for performance already exist or if the goal is merely to demonstrate that the solution is valid, even if it is not the best. For example, Soar learned rules for multiple board games to show benefits of transfer learning (Kirk and Laird, 2016), the SASE robot traveled along the hallway under human supervision (Zhang et al., 2005), and SPA parsed simple commands (Stewart and Eliasmith, 2013).

Performance analysis

Performance analysis focuses on running time and computational resources needed for running the program, both important measures for assessing the model's scalability and practical viability. However, for cognitive architectures, computational complexity and responsiveness are rarely reported because the vast majority of applications and tasks they perform are on a small scale.

Only a handful of examples where performance is of concern focus on speed of execution in time-sensitive domains, such as robotics and human-computer interaction, however the evaluation itself is often limited and informal. For instance, the analysis of search space for rule inference for the Disciple architecture was done only for a single rule (Tecuci and Hieb, 1996). Similarly, reduction in the number of scene captures performed by MDB during object search was shown only for a single scenario (Bellas and Duro, 2003). Assertion of DIARC's planning time under 1 s even for complex environments was demonstrated with a single sample scenario (Talamadupula et al., 2010). For PRS runtime statistics for three illustrative equipment failure scenarios (Ingrand and Georgeff, 1990). Lastly, effects of input size on the running time of ART-based image classification algorithm were demonstrated for a small set of 26 images representing letters of the English alphabet (Kane and Paquin, 1993).

Human evaluation

Human assessment involves asking human subjects to use the system and report on their experience. Naturally, human evaluation is most suitable for the applications that involve human-machine interaction (HMI). Although there are many cognitive architectures that have an HMI component, only some of them have been evaluated with human participants—Disciple, Kismet, TCA, and CORTEX. Disciple is an architecture for knowledge-based agents that collaborate with and learn from domain experts, some in safety-critical domains, such as combat operations and emergency response. Disciple's evaluation focused on the usefulness and correctness of the suggestions made by

the system, and the ease with which mistakes could be corrected. Kismet, a social baby-like robot, was evaluated for its social abilities. The salesman robot Gualzru and the office robot Xavier controlled by CORTEX and TCA, respectively, were tested for ease of interaction with the users and for task completion. A summary of human evaluation of these architectures follow:

- *Expert vs. non-expert users.* Disciple was designed for decision-critical domains, such as combat operations and emergency response, therefore it was evaluated by experts. The rest of the systems were meant for the general public and therefore evaluated by non-expert subjects not involved in the development.
- *Time and number of interactions* were not reported, except for Kismet, where individual experiments were conducted once and lasted 20–30 minutes (Breazeal et al., 2001). Experiments with CORTEX took place during the course of a day, Disciple over several days, and the TCA experiment ran for a few hours daily over 3 years (1995–1998), therefore many users could potentially interact with the system multiple times.
- *The number of subjects varied.* Kismet was evaluated with 4–5 users, Disciple with 20, and CORTEX with 50. TCA did not report the number of individual users, however, their experiment was accessible to anyone with internet access and over 30,000 tasks were placed over the course of the experiment.
- *Feedback modality.* Evaluations of Disciple and CORTEX relied on surveys with categorical scales. The users were asked 10–40 specific questions regarding their interaction with the system and its outputs. On the other hand, the subjects interacting with Kismet were not instructed. Instead, their interactions with the robot were observed by the researchers and recorded, but no other feedback was requested. TCA developers likewise relied on recordings of interactions and less so on informal feedback provided via online guestbook.
- *Analysis of responses* was largely limited to descriptive statistics for survey data (CORTEX, Disciple) and task completion rates (TCA). No statistical analysis was performed. In addition, in all cases, a qualitative summary was provided to describe the positive and negative outcomes of evaluation.
- *Comparative evaluation* is usually not done, i.e. human users interact with a particular version of a single cognitive architecture and provide feedback individually.

Despite the lack of common procedure for human assessment of cognitive architectures, even these informal tests provide useful feedback and expose unanticipated limitations. In all cases described above, users outside the development team interacted with the respective systems in unexpected ways, revealing bugs and deficiencies. For example, the Xavier delivery robot crashed on the first day of operation because the users were telling it to go to the location it already was at (Simmons et al., 2002). The salesman robot Gualzru experienced difficulties hearing in noisy environments. People who wanted to interact with the robot had to move closer to it, which put them outside the range of the robot's stereo camera used for person identification. Based on additional user feedback, the robot was reprogrammed to face the person

talking to it and not to follow people who declined its request for conversation (Romero-Garcés et al., 2015a).

10.1.2 Comparative evaluation

Up until this point, we have considered only individual evaluations, i.e. testing a single architecture on a set of test cases. Comparative quantitative evaluations use the same strategies and measures as individual ones but apply them to more than one entity. There are three aspects of comparative evaluations that we will discuss: the models comparison set, test data, and purpose of evaluation.

Model comparison set. There are several options to use for comparisons: baselines, variations of the same model, models based on other cognitive architectures, and specialized narrow models. Three-quarters of all comparative case studies are ablative, i.e. the comparison is made to a version of the model based on the same architecture. There are almost no comparisons against the models based on other cognitive architectures with very few exceptions, e.g. ACT-R was compared to EPIC (Byrne et al., 1999) and EPIC was compared to GOMS (Kieras, 2005). As an alternative, cognitive architectures are sometimes compared to specialized AI or cognitive models. Finally, the scope of comparisons is minimal: ablation studies may consider 4–5 variants[2] but for comparisons to other works usually only 1–2 models are used. Finally, the use of baselines for evaluation is surprisingly limited, especially given the difficulties with making other types of comparisons.

Test data. Virtually all cognitive architectures we reviewed have been evaluated on small sets of hand-picked test cases that are selected for each application. Partly, this is an artifact of their longevity. Small-scale custom evaluation was common in most of the history of AI, and only in the past decade become more standardized. Currently, benchmarks serve as one of the main evaluation methods in AI for assessing model performance. A benchmark is typically a large and diverse set of data points corresponding to a specific task or tasks, with defined ground truth, metrics, and procedures for evaluation. To enable comparisons, most benchmarks are either publicly available or provide alternative means for evaluation, for example, via a public API.

Benchmark-based evaluations of cognitive architectures are not common, comprising less than 3% of all evaluations. Some notable examples are listed below.

- OSCAR-based model of deduction was tested on a large subset of the TPTP library (Sutcliffe and Suttner, 1998) of test problems for automated theorem-proving (Pollock, 1999);
- ACT-R model of backgammon was evaluated against backgammon pubeval (benchmark player) (Sanner et al., 2000);
- A relational learning system based on PRODIGY was applied to planning problems at AIPS'00 (Aler et al., 2003);
- Bayesian ARTMAP model of image classification was tested on 20 image datasets (Vigdor and Lerner, 2007), including the USPS dataset of handwritten text (Hull, 1994);

[2]As an exception, Marewski and Mehlhorn (2011) built 39 ACT-R models with different assumptions and tested then on a battery of human

- Companion model of analogical question answering was applied to a challenging text comprehension dataset ProPara[3] (Ribeiro et al., 2019).

Purpose of evaluation. In 80% of the cases, evaluation focuses on effectiveness, i.e. how well the task is accomplished, rather than time or space complexity of the solution. Among the one-fifth of the studies that consider runtime performance, the majority target optimization. Sometimes, performance complements efficiency evaluation. For example, the ACT-R model of backgammon won fewer games than the classic TD-Gammon algorithm but required orders of magnitude fewer samples to train (Sanner et al., 2000).

The advantages of comparative evaluations are obvious: they rank different solutions, help identify failures, and track progress toward solving a problem. In addition, the mere process of evaluating others' work has inherent benefits. By doing so, researchers are compelled to closely examine alternative solutions, which in turn helps them find gaps in their own understanding and gain a deeper insight into the problem at hand. Thus, comparative evaluations promote discussion and exchange of ideas and solutions. Despite its evident benefits, comparative evaluation of cognitive architectures has not become a dominant or even significant assessment method for a combination of the following reasons:

- *Level and granularity mismatch.* Cognitive architectures come from different backgrounds and model phenomena on different levels and with varying granularity. These factors do not prevent comparisons outright, but they do make them more difficult.
- *Large number of phenomena to investigate.* The number of human cognitive abilities is vast. As a result, even if two cognitive architectures study similar phenomena, the number of concrete tasks to consider is large enough that the overlap between them is small.
- *No other models available.* Consequently, if the task has not been investigated before, there is nothing to compare to. This was often the case for the early architectures, but less so in many of the current projects.
- *Missing software infrastructure.* For most cognitive architectures, the software is not available, so it is often impossible to test them on a new task for comparison. Even if the code or binary is available, it may not be possible to run due to its complexity and lack of documentation.
- *Inadequate performance.* Comparisons on modern benchmarks may not be favorable; cognitive architectures historically performed on par or worse than narrow AI approaches whenever such comparisons have been attempted. As a result, there is less incentive to conduct such studies.
- *Lack of expertise.* Expert knowledge of multiple cognitive architectures is needed to conduct the evaluation fairly. However, few researchers have such expertise.
- *Low interest.* Rarity of quantitative comparisons in the literature and personal communication with some practitioners point to a lack of interest from the community in this type of assessment. Even though data and narrow AI models are now widely available for a number of tasks, particularly in the vision and language domains, cognitive architectures are rarely tested against them.

[3]https://allenai.org/data/propara

To be fair, comparative methodology should not be treated as the final goal. The shift toward benchmarks in AI during the past decade plus demonstrated progress mainly on the applied side, whereas basic understanding of intelligence has been far less affected. The cognitive architecture community is addressing problems whose inherent solution is far larger than modern applied AI research and far less amenable to brute-force approaches facilitated by mass deployment of GPUs (see Chapter 11). Further, the desire to be truly explanatory of human intelligence is approached with greater respect for human data than AI at large. Because performing comparative comparisons properly is difficult, fewer of them are attempted. The solution moving forward will require input from the entire community to develop appropriate methodologies. The next section will discuss potential directions for further investigation.

10.2 Ability-based evaluation

Ability-based evaluation is meant to be more high-level, broad, and holistic than a task-based one. The latter, as we saw in the previous section, is already non-trivial but determining whether the agent has a certain ability or even intelligence is a different matter altogether. Not because there are no definitions for cognitive abilities or intelligence and the methods for measuring them, but because there are too many. Therefore, any declaration of ability or intelligence is bound to be context-dependent and lacking in some respects, and hence more likely to be contested on those grounds. Nevertheless, evaluating intelligence of artificial agents (including cognitive architectures) has undoubtedly been one of the drivers of progress in the field. In this section, we will look at two directions toward this goal: 1) Turing test and alternative tasks for measuring intelligence and 2) evaluating cognitive and biological plausibility.

10.2.1 Turing test and beyond

The Turing test, which predates the field of AI itself, was meant as a simple and intuitive way of testing whether a machine can think. Although its value as a test of intelligence has been questioned, it is an interesting and thought-provoking concept that remains a part of the ongoing discussion regarding the nature of human and artificial intelligence.

Original and modified Turing test

The Turing test is named after Turing (1950) and is based on the following imitation game with three human participants: a man (A), a woman (B), and a judge (C). The goal of the judge is to identify which one of the participants is male and which is female. The goal of the man is to imitate the woman, and the role of the woman is to help the interrogator. To obscure their identities from the judge, participants A and B are located in a different room and communicate with the judge using a teleprinter. However, when Turing adapted the imitation game for testing machine intelligence, due to the somewhat ambiguous writing, two interpretations emerged:

The original Turing test. According to the literal reading of the 1950 paper, human participant A is replaced with the machine and the rest of the

setup remains as it was, i.e. the conversation is between three participants: A—a machine impersonating a woman, B—a man, and C—a judge, who decides which of A and B is a man or a woman.

The modified Turing test. A more familiar version of the Turing test is the game with two participants: a judge and an entity that may or may not be human. The judge has to decide whether a machine or a real person is answering the questions. The goal of the machine is to fool a judge into believing it is human. This reading is sometimes referred to as standard (Piccinini, 2000) and is consistent with the Turing's own statements given in a 1952 radio interview (Copeland, 2004, p. 495). The advantage of this version of the test is that it is easier to administer and arguably easier to pass for machines.

Although the modified version of the Turing test (sometimes referred to as "species test") is dominant in the literature and the one implemented in practice, Pinar Saygin et al. (2000), Traiger (2000), and Sterrett (2000) argue that the original setup ("gender test") is a more fair and unbiased test of intelligence. Since the test requires both the machine and a person to imitate someone else, it leads to a more nuanced comparison between the two. For example, a machine can be more successful at impersonation than another human, an outcome not possible when only the machine has to convince a judge of its humanness, while the other participant acts as themselves. Another advantage of the original formulation is that it is less dependent on the skill of the interrogator, since both participants will be subject to the same evaluation. Similarly, the judge's beliefs about machine behavior become irrelevant, potentially reducing bias in their assessment. Finally, the original test is more complex, as three entities must interact, instead of two.

Turing test in practice

In 1950, Turing predicted that in fifty years a program of about 100 megabytes[4] will play the imitation game well enough so that an average human interrogator will have only 70% chance of identifying it correctly after a five-minute interview (Turing, 1950). In the 1952 BBC interview, Turing extended the prediction to a hundred years.

The test of this prediction came ahead of schedule, in 1991, when the Loebner Prize was established. The first ever winner, a chatbot named PC Therapist III (Weintraub, 1992), convinced 5 out of 10 judges that it was a human (Epstein, 1992a). Two other notable chatbots, Cleverbot and Eugene Goostman, also claimed to have passed the Turing test. Cleverbot was judged in the 2011 Techniche festival in India as human by 59.3% of those who interacted with it (Aron, 2011). Three years later, at the test held at the Royal Society of London, a chatbot convinced 33% of the judges who talked to it that it was a Ukrainian teenager named Eugene Goostman (Warwick and Shah, 2016).

Although the three chatbots above and many others technically satisfied the conditions indicated by Turing, whether they actually passed the test is debatable. Statistically, the sample of conversations was too small to be significant, and reading the transcripts raises additional questions. Anecdotally, generic

[4]Turing's (1950) paper mentions storage capacity of 10^9 without specifying the units. We assume he meant bits, since bytes were introduced after his death.

conversations were in favor of the machines, whereas strictly defined topics disadvantaged human participants. As Epstein (1992a) notes, during the first Turing competition, the out-of-context remarks made by PC Therapist III were perceived as fitting the "whimsical conversation" topic that its terminal was labeled with. At the same time, three judges penalized a human participant, Cynthia Clay, who chose the topic of "Shakespeare" because her depth of knowledge was deemed unnatural. The judges with expertise in computer science and familiarity with the chatbot technology also had an easier time spotting response patterns and errors typical of machines (Shieber, 1994). Analysis of the transcripts by Halpern (2006) points to flawed approaches to conversation from some judges who insisted on short answers and keeping the conversation on point to the detriment of human participants.

Despite improvements in the chatbot technology, the subsequent winners of the Loebner Prize competition received lower scores and none won the silver or gold medal for convincing more than half of the judges and passing the additional tests of text comprehension, visual, and auditory input, respectively.[5]

Criticisms and objections

By many accounts, the Turing test had lost its status as a grand challenge in AI at the end of the 20th century but remained a starting point for discussions on measuring intelligence. Before we move toward describing the test's shortcomings, we believe it is necessary to point out the good qualities of the original proposal: the test is easy to administer, entry cost is low as it requires no sophisticated equipment, the results are clear to obtain and explain, the task is challenging and interesting enough for researchers to work on, and simple enough to appeal to the general public.

Nevertheless, shortly after publication, Turing's idea was met with criticisms of a both philosophical and practical nature. Given the volume of commentary that has accumulated over the past seventy years, below we summarize only the core issues with the test.

Behaviorism. The formulation of the test was intentionally abstracted from *how* the artificial agent solves the task, which can be seen as both its strength and weakness. On one hand, it is a way to avoid debating the definition of intelligence. On the other, focus only on the final outcome ignores many aspects of intelligent behavior, such as how one perceives anything, how one decides what is important, how one decides what actions to take, and more.

Context-dependence. The context and interface through which the communication occurs may alter the outcome of the test. During one informal session, a student acted as a machine and the other subjects did not suspect they were talking to an actual human, even though the student did not change their behavior or try to be more machine-like (Watt, 1996).

[5]Regrettably, the Loebner Prize became defunct circa 2020 before large-scale language models (LLMs) could participate. Although the new crop of neural models exceeds Turing's conditions size-wise, they are arguably better than their predecessors at generating coherent and grammatically correct text. However, besides the early reports of the language model LaMDA fooling a Google engineer that it was sentient (Lemoine, 2022), the majority of LLMs did not come close to passing for a human. To list a few, ChatGPT, arguably the most advanced system optimized for conversation, failed to convince a philosopher (Hanna, 2023), volunteers in a small study (Guo et al., 2023), and marketing experts (Cook, 2023).

Pass/fail scoring. The all or nothing evaluation is not suitable for research for the following reasons:

- *It is not diagnostic.* A single fail/pass score is not helpful, because it does not reveal what elements of the system need improvements.
- *It gives no partial credit.* An incomplete solution or a solution that does well on some aspects of the task is not rewarded.
- *It is not incremental.* Gradual development of the solution is discouraged since the task is not decomposable and its difficulty cannot be controlled.

Passing the test ≠ intelligence. This objection questions whether the test is a sufficient medium for demonstrating intelligence, given that obviously intelligent agents (humans) have failed the test in the past (Hofstadter, 1982) and that there are possibilities for obviously non-intelligent agents to game the test to pass it. Several such agents have been discussed:

- *Blockhead*—A giant look-up table or decision tree for every possible question (Block, 1981) (an argument similar to Searle's (1980) Chinese Room).
- *Random*—A random finite state automaton generating random English sentences may get lucky and fool the judge (Pinar Saygin et al., 2000).
- *Artificial stupidity*–An uncooperative agent that evades interaction but nonetheless responds in a human-like manner (Hutchens, 1997).

Biased judging. The need for a human interrogator introduces additional issues, such as:

- *Difficulty automating and scaling the test.* Having human judges limits the number of tests that can be administered.
- *Dependency on the skill of the interrogator.* Depending on who the judge is, the questions may be of arbitrary difficulty.
- *Possible advantages to human participants.* Any prior knowledge about the participants can be exploited by the judge.

Aviation analogy. Some argue that the test of human-like intelligence is not necessary or is even harmful, as it will lead to attempting to make a literal copy of the human mind. For example, airplanes are designed to fly without emulating all biological principles of flight. The result is a different kind of flight, superior in some aspects and inferior in others. By analogy, we should put aside trying to imitate human intelligence in every aspect and accept that machine intelligence may be qualitatively different.

Anthropomorphism. The Turing test has been called chauvinistic because it was modeled on human intelligence to the disadvantage of other intelligent entities, e.g. those that cannot communicate using language or do not conform to human cultural norms.

Many of these objections can be deflected if one does not believe that Turing's imitation game was ever meant to be a test or working definition of intelligence. Copeland (2000) and Berrar et al. (2013) reach this conclusion after a thorough analysis of Turing's papers and interviews. Turing used the terms "game" and "test" interchangeably to describe the evaluation procedure and at no point precisely defined the meaning of "thinking machine," suggesting that the test was a speculation on his part rather than a rigorous proposal.

Despite its deficiencies the Turing test is still part of the discourse around intelligence. While the original test has been largely abandoned as a real

goal for developing AI, it continues to fuel discussion around assessing human and non-human intelligence. Every new proposal aims to overcome the shortcomings of the Turing test and its variations by 1) modifying the task and evaluation to make the test more robust against cheating and conductive to incremental research and/or 2) extending the test beyond the linguistic domain toward testing other aspects of intelligence.

Turing test variations

Variations of the Turing test aimed at improving the original proposal in various ways, mainly to address the possibility of gaming it. To assess other aspects of intelligence, tests that involve embodiment and more complex tasks were also proposed. Below we list some of them.

Linguistic Turing tests. Most of these tests are still administered in the same way as the original proposal, i.e. there is a person who judges how successfully the machine was able to impersonate a human.

- *Inverted Turing test.* Watt (1996) suggested an Inverted Turing test, where the task of a machine is to tell people from machines. The premise is that to do well on this task, the machine needs to develop the same understanding of naive psychology as human judges.
- *The Feigenbaum test.* The Feigenbaum test uses the Turing test framework to evaluate deep domain expertise. First, up to ten research areas are selected, such as astrophysics, computer science, molecular biology. For each area, two elite scientists are chosen—one as an opponent to the machine and one as a judge. The judge then interviews the machine and the scientist in their area of expertise and decides which of them is a human scholar (Feigenbaum, 2003).
- *Moral Turing Test.* Another variation proposed by Madl and Franklin (2015) keeps the standard Turing test framework but evaluates the morality of the agent by constraining the topic of conversation.
- *Questioning Turing Test (QTT).* Damassino (2020) proposed switching the task from question answering to question asking. Now instead of waiting for the judge's input, the machine interrogates the judge, for example, by playing a twenty questions game. Potentially, the task can be extended to First Aid QTT (query the judge acting as a patient on their medical history) or Detective QTT (interrogating a judge pretending to be a suspect). The new format prevents gaming and discourages brute-force agents by placing a limit on the number of questions asked.
- *Handy Andy.* A test devised by Cohen (2005) requires a machine to produce a five-page report on an arbitrary topic with the tools and resources available to humans, i.e. access to the internet, encyclopedias, etc.
- *Text comprehension.* To pass, a machine needs to answer an open-ended question about a given text or potentially any other source of information, such as a podcast, or a video (Paritosh and Marcus, 2016; Kočiskỳ et al., 2018).

Robotic Turing tests. A Total Turing Test (TTT) (Harnad, 1991) reframed the Turing test as a robotic challenge that required the systems to be embodied and perform real tasks in the real world on par with humans. TTT was a thought experiment primarily to defeat Searle's argument and demonstrate the limitations of text-based evaluation, but at least two incarnations of the

embodied Turing test exist. One is RoboCup (Mackworth, 1993), a well-known challenge with the goal to beat human world champion soccer teams by the year 2050. The second challenge similar in spirit is proposed by Ortiz Jr (2016) and features everyday tasks, such as setting up tents, building modular furniture based on instructions, and real-world collaborative interaction.

General game playing. Board games and later computer games have been an invaluable tool for developing AI. Therefore, it is not surprising to see many adaptations of the Turing test to the task of playing games. One of the early proposals is "game player's Turing test" (Epstein, 1994). AI passes this test if it is reliable (can win consistently against players of any skill level) and powerful (can exploit the errors of the opponent). More literal renditions of the Turing test for games are the 2K BotPrize (based on a first-person shooter Unreal Tournament 2004) (Hingston, 2009), the Turing test track for Super Mario AI benchmark (Togelius et al., 2013), and the Turing Tournament (two-player board games) (Arifovic, 2005). The first two rely on human judges to determine whether the AI playing the game acts like a human player. During the Super Mario AI challenge, an audience of non-experts viewed videos of humans and agents playing the game and cast their votes. Initially, 2K BotPrize also asked external observers to judge the humanness of the game agents. In the next iteration of the test, the judges played the game against human and artificial opponents to decide which one is which (Hingston, 2010). The Turing Tournament substitutes human arbiters with computer algorithms, therefore two types of programs take part in the contest—emulators that aim to mimic human players and detectors that try to tell human players (represented by a prerecorded dataset) from machines.

Miscellaneous tasks. A multitude of tests have been proposed for nearly every human activity. Perhaps, the most known is CAPTCHA, which stands for Completely Automated Public Turing Test to Tell Computers and Humans Apart (Von Ahn et al., 2004). This test is designed to distinguish humans and computers apart by asking them to decipher noisy images or text. Among other proposals are the employment test (passing certification for a regular job) (Nilsson, 2005), the navigation test (navigating a 3D virtual space as well as a human) (Devlin et al., 2021), driving test (driving a car like a human) (Stanton et al., 2020), the non-verbal Turing test (human-like gaze during social interaction) (Pfeiffer et al., 2011), graph drawing (compares graph layouts generated by machines and humans) (Purchase et al., 2020), and even artistic creativity test (create art indistinguishable from human) (Boden, 2010).

Multi-task benchmarks. I-athlon was one of the first proposals for a multi-dimensional Turing test akin to Olympic Decathlon (Adams et al., 2016). Several candidate events were described, such as image understanding, diagram understanding, speech generation, natural language generation, collaboration, competition, reasoning, creativity, interaction, embodiment, and more. Specific tasks for each event were to be automatically generated and evaluated without any human involvement.

Although I-athlon never materialized, several similar multitask benchmarks have been developed. One example is the Visual Turing Test, which contains a set of images extensively annotated with object locations, properties, and queries regarding what is depicted in the scenes (Geman et al., 2015). While

some queries may be relatively easy, for example detecting whether a certain object is present, others may require spatial reasoning about relationships between objects in the scene. The largest to date multitask benchmark is called BIG-Bench, which stands for Beyond the Imitation Game (Srivastava et al., 2023). Although it is limited to natural language and arithmetic problems, it more than makes up for it by sheer volume of implemented tasks. As of today, more than 200 tasks contributed by over 400 authors are available for anyone to try.[6]

Recent developments in deep learning, especially the onslaught of the large language models (LLMs) and chatbots, revealed two fundamental problems with the linguistic tests as a measure of intelligence: they are susceptible to data contamination (training on benchmarks) and shortcut learning (exploiting biases in the datasets) (Mitchell, 2023). There are solutions that mitigate these issues by introducing subtle changes to test deeper understanding (Dong et al., 2023; Shapira et al., 2024) and curating the data to avoid biases (Sakaguchi et al., 2021), however, these are temporary. Similar problems are also likely to arise in multimodal tasks.

Although foundation models continue to shatter benchmarks, a cursory interaction with the current top models is sufficient to realize that their intelligence is a mirage. Perhaps, the "you know it when you see it" criterion for intelligence of the original Turing test has merit after all.

Psychometric AI

Psychometric AI advocates assessing artificial agents using the tools for measuring human intelligence. Bringsjord and Schimanski (2003), who introduced the term, propose to consider an agent intelligent if it is "capable of at least solid performance on all established, validated tests of intelligence and mental ability, a class of tests that includes not just rather restrictive IQ tests, but also test of artistic and literary creativity, mechanical ability, and so on." The specific tests they mention are Wechsler Adult Intelligent Scale (WAIS) and Torrance Tests of Creative Thinking (TTCT).

The idea of using standard psychometric tests to evaluate AI agents was also brought up in the cognitive architectures community. For example, Newell (1973b) advocated the use of adult IQ tests, naming WAIS and Stanford-Binet as options. Cassimatis and Bignoli (2011) and Goertzel and Yu (2014) suggested a more gradual approach, beginning with simpler tests designed for assessing child development.

Some psychometric tests of cognitive architectures already exist. For example, models based on ART (Carpenter et al., 1990), Companion (Forbus et al., 2011), ACT-R (Peebles, 2019), and SPA (Eliasmith et al., 2016) have been evaluated on Raven's Progressive Matrices (RPM) that test non-verbal intelligence. The Companion architecture has also been applied to a challenging subset of the problems from the Bennett Mechanical Comprehension Test (Klenk et al., 2005).

Although psychometric AI relies on established tests and methodologies, it is susceptible to the same weaknesses as the original Turing test. First, the use of human IQ tests is regarded by many even more anthropomorphic than the imitation game (Hernandez-Orallo, 2000; Legg and Hutter, 2007b; Dowe

[6] https://github.com/google/BIG-bench

and Hernández-Orallo, 2014; Besold et al., 2015). Second, solving a battery of intelligence tests can be gamed as easily as convincing a human judge. In fact, a Perl program consisting of only 960 lines encoding heuristics for common types of questions received an average score of 99.6% across multiple IQ tests (Sanghi and Dowe, 2003).

Universal psychometrics

For completeness, we also briefly mention universal psychometrics (Dowe and Hernández-Orallo, 2014), which aims to address the anthropomorphism of the Turing test by providing a universal definition and measure of intelligence. Advocates of this method follow Legg and Hutter's (2007b) definition of intelligence as "the ability to achieve goals in a wide range of environments." The interaction between the agent and the environment is framed as reinforcement learning, a measure of intelligence then can be defined as aggregate performance across a set of Turing computable environments with respect to Kolmogorov complexity of the agents, rewards, and environments (Dobrev, 2005; Legg and Hutter, 2007b; Hernández-Orallo and Dowe, 2010). The claimed advantage of this metric relative to the Turing test is that it is formally motivated, precisely defined, and can be applied universally, anytime, and adapted to testing specific cognitive abilities by changing the set of environments.

Besides the issues with this understanding of intelligence discussed earlier in Section 1.2, it is also difficult to apply in practice. Selecting a truly unbiased set of tasks and environments from a universal distribution is one of the problems. However, the most significant challenge is developing an interface for specific types of agents (e.g. rewards for animals and verbal instructions for adults) without giving them the advantage over others (Dowe and Hernández-Orallo, 2014). Unsurprisingly, the first (and only) attempt at applying a version of the test to humans and a simple reinforcement learning agent was largely inconclusive (Insa-Cabrera et al., 2012), whether due to the questionable nature of the initial assumptions or due to the simplicity of the task itself. In all, the experiment demonstrated that the practical limitations of the test outweigh the benefits of theoretical rigor, suggesting that universal psychometrics may be ill-suited for widespread use in its current form.

10.2.2 Cognitive and biological plausibility

The approaches we have discussed so far focus mainly on behavioral similarity between machines and humans (or other types of intelligence). Just like in the original Turing test, evaluation is purely functional, placing no restrictions on the implementation. In other words, as long as a passing score is achieved, how the task is completed does not matter, even if the solution is a giant look-up table (Legg and Hutter, 2007b).

This is in stark contrast with the goal of cognitive architectures, which includes achieving cognitive and biological plausibility of the internal mechanisms as well. Perhaps, it is one of the reasons why so few cognitive architectures have been evaluated on Turing-like tests. The only example we are aware of is the CERA-CRANIUM bot for playing a video game that was judged as a human player by 30% of observers during the 2K BotPrize in 2010 (Arrabales et al., 2009a).

Test-based evaluation

To address the behaviorism of the Turing test Harnad (1991) proposed a new extension of the Total Turing Test mentioned earlier, which was named the Total Total Turing Test and introduced constraints on the implementation. Specifically, it required that the machine should not only be embodied but also indistinguishable from humans down to the level of neurons and molecules. This is not to be confused with the Truly Total Turing Test (Schweizer, 1998), which called for incorporating a full range of human behavioral data for assessment. Regrettably, neither proposal mentioned concrete tasks or evaluation methods.

It took two more decades for another approach to emerge. The Cognitive Decathlon was proposed for the initial stage of the BICA program sponsored by DARPA (Mueller et al., 2007). Although the program itself was terminated prematurely, work on the Cognitive Decathlon continued. The authors started by reviewing hundreds of psychological studies to select the tasks that had the following properties: a) human performance on the task was well understood, b) there was a computational model, c) the task was related to the core abilities of a 2-year-old child, and d) the task was a component of a more complex task or ability. Selected tasks were grouped into broad categories covering vision, search, manual control, and learning, but not reading, writing, or numerical skills. Assessment considered neurosimilitude (adherence of design and computation to characteristics of biological systems), task-specific evaluation (comparisons of model activity and output to results in the literature), and direct human data comparisons (between model internal activity and fMRI data from human subjects performing the same tasks). Even though a partial implementation of the Decathlon exists (Mueller, 2010), we could not find any cognitive architectures that were evaluated.

The only practical evaluation of multiple cognitive architectures on human data to date was conducted as part of the Agent-Based Modeling and Behavior Representation (AMBR) project led by the Air Force Research Laboratory (AFRL) (Gluck and Pew, 2006). The challenge consisted of two tasks: a simplified air traffic control task and a category learning task from the classic study by Shepard et al. (1961). Four cognitive architectures were tested, among them ACT-R, COGNET, DCOG, and EASE.[7] The models for each task were built by the developers of the cognitive architectures and then evaluated on the human data to test how well they predict various performance measures on multiple tasks at multiple levels of aggregation and under different conditions. It was, however, difficult to rank the models because each had limitations and fit only parts of the data better than others.

Analytical evaluation

Due to lack of widely available test banks and agreed upon evaluation methodology, analytical evaluation remains one of the most common comparative evaluation methods in the literature. Analytical evaluation starts with a set of cognitive architectures and a set of broadly outlined cognitive abilities, e.g.

[7]ACT-R and COGNET are covered in this book, whereas DCOG and EASE did not satisfy the criteria listed in Section 1.6. DCOG stands for Distributed Cognition Framework (Eggleston et al., 2000) and EASE is an abbreviation of Elements of ACT-R, Soar, and EPIC (Chong, 2004). EASE is a hybrid system extending another hybrid

learning, autonomy, perception, etc. The next step is to describe components of the cognitive architectures and past works that are relevant for each ability. Sometimes, based on these descriptions, a categorical score is assigned that indicates whether the ability is present and (optionally) to what extent.

- (Wray et al., 1992)[8]—an early comparative survey of 12 cognitive and agent architectures. Considers the following properties and capabilities: organization, memory and knowledge representation, learning, planning, problem-solving, performance, perception, and interaction with the environment;
- (Pew and Mavor, 1998)—a descriptive analysis and comparison of 15 architectures based on their original purpose, submodels (perception, memory, motor control), knowledge representation (declarative, procedural), higher-level cognitive functions (learning, planning, decision-making, situation assessment), multitasking (serial/parallel, resource representation, goal management, multiple human modeling), implementation, and output;
- (Anderson and Lebiere, 2003)—a comparison between ACT-R and connectionist approaches against criteria of the proposed Newell test, which include flexible behavior, real-time performance, adaptive behavior, large knowledge base, dynamic behavior, knowledge integration, natural language, consciousness, learning, development, and evolution;
- (Chong et al., 2007)—a functional comparison of 6 cognitive architectures discussing implementations of perception, memory, goals, problem-solving, planning, reasoning/inference, learning, and their relevance to neurobiology;
- (Vernon et al., 2007)—a comparison of 14 architectures on 7 characteristics (paradigm, embodiment, perception, action, anticipation, adaptation, motivation, autonomy);
- (Thórisson and Helgasson, 2012)—a comparison of 9 architectures across 4 dimensions of autonomy, such as real-time performance, resource management, learning, and meta-learning;
- (Samsonovich, 2010)—a comparative table of 26 cognitive architectures across 50+ dimensions indicating support for common components and features, learning, applications, and limitations;[9]
- (Kotseruba and Tsotsos, 2020)—an overview of 80+ cognitive architectures across 7 core cognitive abilities (sensation, perception, memory, learning, reasoning, decision-making, and meta-reasoning);
- (Laird, 2022a)—a detailed comparison between ACT-R and Soar, focusing on the architectural properties and memory (working, procedural, declarative).

Analytical comparisons in the surveys above have the obvious advantage of breadth, as they generally evaluate multiple cognitive abilities of a larger number of diverse cognitive architectures. Currently, the smallest survey in the list above covers more cognitive architectures than any of the existing quantitative tests.

[8]The survey grew out of a class project at the University of Michigan. The original link is no longer active, but is available at `https://web.archive.org/web/20140212220057/http://ai.eecs.umich.edu/cogarch0/`

[9]This is the only study where all descriptions of cognitive architectures were evaluated by their respective authors who filled out a provided template.

10.3 Summary

- The approaches used for evaluating cognitive architectures can be divided into task-based and ability-based. Task-based evaluation comprises qualitative and quantitative methods that assess performance of the systems or their modules on known tasks with known outcomes. Ability-based evaluation is represented by challenges, test batteries, and analytical assessments designed to reveal broadly defined cognitive abilities.
- Task-based evaluations are far more common than ability-based ones. Within task-based evaluation, quantitative methods are more common than qualitative ones (e.g. demos and conceptual comparisons). The majority of cognitive architectures are evaluated individually, with no comparisons to baselines, other cognitive architectures, or specialized models.
- Goodness of fit to human data is one of the most common task-based methods for quantitative evaluation of the psychologically inspired cognitive architectures. Despite evidence that good fit is a necessary, but not sufficient proof of model validity, not many cognitive architectures provide additional evidence for model complexity, generalization, and falsifiability.
- In 80% of cases, task-based evaluations focus on effectiveness, i.e. how well the task is accomplished, rather than efficiency, i.e. time or space complexity. Among studies that consider runtime performance, the majority target optimization. Sometimes, the complexity of the approach is used to complement efficiency evaluation, especially when efficiency is not favorable.
- The Turing test and its variations in the linguistic and other domains is the most known test of intelligence. However, it has been wrought with practical and theoretical issues and has never been applied to cognitive architectures in its original form.
- Quantitative ability-based evaluation has not been applied to most cognitive architectures. Partial implementation of the Cognitive Decathlon and evaluation of a handful of architectures on the air traffic control task are the only two attempts realized in practice. Besides these, analytical evaluation remains the main method for comparing and assessing the abilities of cognitive architectures.

Part IV

WHAT IS NEXT?

In the next two chapters, we will summarize pending issues and point out possible directions for the field moving forward. First, in Chapter 11, we place research on cognitive architectures in the context of modern artificial intelligence (AI), which is currently dominated by deep learning methods. Then, in Chapter 12, we discuss issues that cognitive architectures faced throughout the years and suggest possible ways of resolving them.

11 Cognitive Architectures in the Deep Learning Era

It is hard not to notice that a family of machine learning techniques called deep learning recently took over many areas of artificial intelligence (AI). This trend began in the early 2010s, and since then, both academic and industry research has poured substantial resources into collecting large datasets for training models with an ever-increasing number of parameters.

The successes of deep learning in multiple domains led some to claim that human-level abilities across many tasks have been achieved or surpassed. Additionally, some argue that deep learning is the best model of the human brain currently available and that artificial general intelligence (AGI) may be imminent. This chapter will examine the validity of these claims, investigate the effects of deep learning research on cognitive architectures, and speculate about the future of cognitive architecture research in this new era.

Section 11.1 begins with the definition of the term deep learning and its theoretical roots in connectionism.
Section 11.2 examines the interaction between cognitive science, neuroscience, and deep learning and mutual influences they exert on one another.
Section 11.3 discusses examples of integration of neural networks and biologically inspired learning mechanisms explored in cognitive architectures.
Section 11.4 discusses whether deep learning can potentially result in a cognitive architecture.

11.1 What is deep learning?

The term "deep learning" has become nearly synonymous with AI in academic and popular literature across multiple domains. Although deep learning may seem like a recent phenomenon, its roots run deep in the history of AI. In this section, we will outline the origins of deep learning, discuss the factors that enabled its quick and widespread adoption, and establish its current place within machine learning.

11.1.1 From connectionism to deep learning

Modern deep learning can be traced back to early connectionist models, starting with the model of neuron proposed by McCulloch and Pitts (1943) and Rosenblatt's (1958) perceptron. As discussed earlier in Section 2.1, after the rise of connectionist approaches to modeling cognition in the 1980s, neural networks steadily gained popularity through the 1990s, and began showing promising results in various benchmarks in the 2000s (Schmidhuber, 2013).

Most historical accounts consider 2012 as the watershed moment of the modern deep learning era. That year, the SuperVision team won the ILSVRC image classification challenge based on the ImageNet dataset (Russakovsky et al., 2015). The team used a convolutional neural network (CNN), later named AlexNet after one of its authors (Krizhevsky et al., 2012). The success of AlexNet was attributed to its deep multilayered architecture as well as novel data augmentation and regularization techniques. By 2014, all the top entries in the ILSVRC competition were deep convolutional networks (Russakovsky et al., 2015).

The popularity of deep learning continued to rise over the next decade. Hardware improvements, such as availability of fast GPUs and CPUs, together with increasing storage capacity, facilitated development of larger models trained on bigger datasets. Initial successes in computer vision and natural language processing tasks prompted the expansion of deep learning into new domains. During this period, new model architectures and optimization techniques were introduced to accommodate new types of data and improve training and inference times for practical applications.

As with expert systems before, a significant factor contributing to the renewed interest in neural networks is substantial investment of financial and human resources by academia and, more importantly, large corporations. For example, the two most popular frameworks for building deep learning models are developed and maintained by tech giants—TensorFlow by Google (Abadi et al., 2016) and PyTorch by Meta (Paszke et al., 2019). Availability of code, extensive documentation, numerous tutorials, and free cloud computing services has further lowered the entry requirements for individuals, which in turn led to the creation of hundreds of thousands of projects. Besides pursuing business interests, corporations have also increased their influence on academic research in computer science, AI, and related disciplines. According to recent reviews, industry now leads academic institutions in hiring PhDs, publishing in top conferences, and producing state-of-the-art models (Kotseruba et al., 2021; Ahmed et al., 2023).

11.1.2 Deep learning and machine learning

Deep learning is one of many statistical machine learning techniques that enable computers to learn from data as opposed to manual preprogramming through rules or other hand-crafted representations (Goodfellow et al., 2016; Wiley, 2016; Chollet, 2018; Paluszek and Thomas, 2020). "Deep" in deep learning is not a metaphor for the eponymous concept introduced by the educational psychologist John Biggs (1988) but a literal reference to the number of layers in artificial neural networks (ANNs), also known as multilayer perceptrons (MLPs). While technically any network with two or more layers can be considered deep, typical modern deep networks contains tens or even hundreds of layers. Networks with only a handful of layers are referred to as shallow (Bengio, 2009, p. 6).

All deep learning models have four main components: architecture, training objective, learning rules, and training data. The architecture defines types of computational units and how they are organized. The training objective is a function being optimized with respect to the given task and properties of the training data. Learning rules prescribe how the weights of units are changed

to reach the training objective. The scale and diversity of the training data affect both the training process and its outcomes.

A precise demarcation of deep learning within machine learning is problematic because deep learning has absorbed many methods and continues to evolve at a fast pace. As a result, the term now refers to a much broader range of architectures and learning techniques than the original set that included MLPs, convolutional and recurrent networks, as well as backpropagation and gradient descent. These have now been joined by deep reinforcement learning (Mnih et al., 2015), generative models (Goodfellow et al., 2014), and Transformer-based architectures (Vaswani et al., 2017). In addition, there are numerous examples of classical statistical optimization techniques combined with deep hierarchies and error backpropagation, such as deep boosting (Kuznetsov et al., 2014), deep stacking networks built with support vector machines (Wang et al., 2019), deep decision forests (Zhou and Feng, 2017), and symbolic deep networks (Veksler et al., 2022).

In sum, current deep learning is best understood as a set of learning techniques and representations that are loosely inspired by neural computations but do not commit to a specific theory (Bengio, 2019a; Bengio, 2019b). As a subfield of AI and machine learning, deep learning aims to produce artifacts useful for practical applications and capable of operating at a level surpassing human abilities at that task.

11.2 Cognitive science, neuroscience, and deep learning

The current mainstream deep-learning-based research in AI focuses primarily on real-world applications and relies heavily on large-scale data and engineering. Statistical learning techniques are increasingly being used for extracting knowledge from very large volumes of data to build foundation models (Bommasani et al., 2021) and even automating design of the network architectures themselves, for example, via neural architecture search (Zoph and Le, 2017). In comparison, cognitive science and neuroscience seek mechanistic models that explain how neural computations result in observable behavior. Historically, much of this research has been conducted on hand-crafted models and shallow neural networks.

Despite these differences, there is a connection between cognitive science and neuroscience on one side and deep learning on the other. In this section, we will discuss how knowledge about the human mind and brain has already helped reveal limitations of deep learning models. We will then explore the assertion that they may act as models of the human brain.

11.2.1 Cognitive and biological plausibility of deep learning models

Although computational units in ANNs are commonly referred to as "neurons," it is generally acknowledged that deep learning is based on a very simplified view of neural computation. Below we will present the main arguments against biological plausibility of neural networks and ongoing research toward infusing more biological detail in neural networks for both practical gains

and improvements in the cognitive and biological fidelity of the deep learning models.

Models of neurons

Deep learning networks are composed of many identical computational units, each of which calculates a weighted sum of its inputs and passes the result through a non-linear function that outputs a continuous value (da Silva et al., 2017). The precursor of this model was born out of a collaboration between psychiatrist McCulloch and logician Pitts, who developed a logical engine that associated logical propositions with neuron activity (Abraham, 2002).

The McCulloch-Pitts (MCP) model incorporated most of what was known about the brain in the 1940s, namely that the nervous systems consisted of neurons, that each neuron had a soma (central body) and axon (an elongated portion of the neuron), that synapses connected the axon of one neuron to the soma of another, and that each neuron had a threshold for activation, which spread from the axon terminals to somata. Additional assumptions were necessary for logical calculus, such as the on-off activity of the neurons and the immutability of the network structure (McCulloch and Pitts, 1943).

The MCP model is simple and computationally efficient precisely because it abstracts away much of the messy empirical reality; biological neurons are analog and densely connected (Hopfield and Tank, 1986), they do not update synchronously, do not all have the same fixed delay, and produce sequences of pulses instead of a single value (Hertz et al., 2018). The continuous activation function (e.g. sigmoid) found in modern neural networks was added later to enable analog input and output, as well as learning through gradient descent, but it is also problematic biologically (Maass, 1997). Furthermore, all deep learning networks operate with at most a few different types of neurons that are organized in simple layers, whereas their biological counterparts are much more diverse. By now, over a hundred types of neuronal and non-neuronal cells have been identified in the mammalian brain based on their anatomical properties, location in the nervous system, neurotransmitters they release, and connectivity (Zeng, 2022).

More biologically accurate models incorporate neural dynamics, i.e. they describe the output of a neuron as a series of action potentials (spikes) rather than a real value that changes in discrete time steps. There is a range of spiking neuron models that include varying levels of biological detail (Izhikevich, 2004). The leaky integrate-and-fire (LIF) model introduced over a century ago by Lapicque (1907) is one of the simplest. Similarly to the MCP, it takes the sum of weighted inputs, but the input is integrated over time, simulating a resistor-capacitor circuit. The spike is produced whenever the integrated value exceeds a predefined threshold. Biophysical models, such as the classic Hodgkin-Huxley (1952) neuron model, describe action potential in great detail at the molecular, cellular, and circuit levels. However, the disadvantage of most spiking neurons models is that they are not analytically tractable and therefore cannot be easily applied on a sufficiently large scale (Abbott and Kepler, 2005).

Learning

Information in the biological brain is captured in the strengths of the connections (synapses) between the neurons (Crick, 1989). The main mechanism for forming new memories, adaptation, and skill acquisition is adjusting the

synaptic strengths between neurons, also referred to as synaptic plasticity (Kennedy, 2016).

In computational neural networks, an equivalent of synaptic strengths are parameters or weights between computational units. While the existence and nature of information stored in the human brain at birth are still being debated (Bateson and Mameli, 2007), neural networks are always created "blank"—with weights set to small random values. During learning, the information in the network is updated via gradual weight adjustment. In deep learning architectures, this typically involves backpropagation (backprop) and stochastic gradient descent (SGD). Backpropagation is a technique for computing the gradient of the given cost function with respect to the network parameters, while SGD is an iterative method for updating the weights layer by layer in a backward pass. In the literature, the term "backpropagation" often refers to both of these processes.

A natural question to ask is whether anything similar occurs during learning in the brain. The answer given by Rumelhart et al. (1986a) who popularized backpropagation is that "in its current form, [it] is not a plausible model of learning in brains." The same observation was made by Grossberg (1987) and Crick (1989) and, nearly forty years later, by Whittington and Bogacz (2019) and Hinton (2022). Persistent problems with backpropagation as a model of learning in the brain can be summarized as follows:

Symmetry of forward and backward weights. Backpropagation relies on the same weights for the feedforward inference pass and for the backward error-propagation path. For this to happen in the brain, there should be identical and synchronized bidirectional connections between neurons. However, there is no evidence of such connectivity in the brain or one-to- one correspondences between synapses. While bidirectional connectivity is certainly present in the brain, it does not exist for every neuron and is not necessarily symmetrical.

Weight updating strategy. In the typical feedforward networks, information is computed in a forward pass, whereas backpropagation proceeds in the opposite direction. Thus, each weight is determined based on the global activity of all neurons before it. In the brain, however, the connection strengths are updated locally and there is no evidence of the specific error signal that drives this process.

External control of learning. In contrast to biological systems, artificial learning through backpropagation is not autonomous. An external control signal is needed to switch between learning and inference phases and to decide when to stop learning.

Biological implausibility and practical success of backpropagation may seem contradictory and attempts have been made to resolve this conflict, for example, looking for backprop-like signals in the brain (Lillicrap et al., 2020) or applying backpropagation to more biologically accurate spiking neuron models (Pfeiffer and Pfeil, 2018). Even more efforts were dedicated to addressing the issues with the method itself. Among recent proposals are backpropagation through random connections (Lillicrap et al., 2016) to remove the need for symmetric weights, equilibrium propagation (Scellier and Bengio, 2017), and the forward-forward algorithm (Hinton, 2022) that do not rely on separate circuits for error propagation, and continuous updates (Bengio et al., 2017) to ease

the requirement of external supervision. However, none of these approaches fully address all problems with backpropagation or match its performance on common benchmarks (Bartunov et al., 2018).

Generalization

Generalization is arguably the most important property of learning. Simply put, it is a measure of how well a biological or artificial entity can deal with inputs that have not been experienced before. Generalization is assessed in two stages. In the first stage, the entity is trained to respond to a sample of stimuli drawn from some data-generating process. Once proficiency is reached on the training data, the second stage begins, where the entity is presented with a set of different stimuli drawn from the same process. Successful generalization occurs when responses to both sets of data are the same.

In the animal kingdom, generalization is ubiquitous and appears in many contexts, suggesting that it is a fundamental characteristic of living organisms (Ghirlanda and Enquist, 2003). While there is no single theory of generalization yet, there are several narrow models that predict generalizability to specific stimuli or in specific tasks (Taylor et al., 2021). In all of them, similarity is recognized as one of the key factors, which includes both physical resemblance and similar representation in the brain. The latter enables generalization between distinct categories of stimuli.

The practical success of deep neural networks is often taken as a sign of their generalizability, but there is no theory that explains exactly how it occurs. One of the main questions is how overparametrized models that are capable of memorizing the entire training set can generalize at all (Zhang et al., 2021; Chatterjee and Zielinski, 2022). Since generalization depends on many factors, such as the size of the training data, size of the input domain, number of weights and layers in the network, and properties of the weights, stability of learning to perturbations in training data, and robustness of performance, there are no analytical bounds that are applicable to an arbitrary deep network (Jakubovitz et al., 2019).

An assumption is often made that random variables in the training and test data are independent and sampled from the same probability distribution (referred to as the assumption of independent and identical distribution or simply IID). The IID assumption is necessary for the tractability of statistical analysis methods, however it is violated in practically all real domains (Cao, 2022). For most, the distribution of samples is unknown and even for small known domains it is prohibitively expensive to sample from in any principled fashion. For instance, sampling from a set of 520 basic LEGO bricks already presents challenges. To cover all possible orientations, poses, and colors of this set would require taking more than fifteen million images, that is before considering varying the backgrounds (Tsotsos and Luo, 2021).

Parallel to theoretical investigations, a lot of experimental work is being done to empirically probe machine generalization. A common theme is applying various distortions or perturbations to the input, to which most deep learning networks are not immune. For example, the performance of image classification models drops precipitously when even modest amounts of noise are added to the input (Geirhos et al., 2018), or when pictures of objects are taken from non-standard viewpoints (Barbu et al., 2019). In a video game setting, adding rectangles or lines to the background causes a complete failure

of neural networks that previously achieved higher scores than human players (Gamrian and Goldberg, 2019). In the natural language domain, deep models fail to generalize to pairs of words not observed together during training (Hupkes et al., 2020) and often cannot incorporate newly learned words in sentences (Lake and Baroni, 2018).

In comparison, humans and other organisms are much less affected by both trivial and more substantial changes to their inputs. Even species with very modest processing capabilities can generalize well beyond physical similarity of stimuli (Ghirlanda and Enquist, 2003). The leading hypothesis is that a small set of innate abilities rather than accumulation of samples through experience enables such generalization (Zador, 2019). This explains why many animal species can function well shortly after being born. Lake et al. (2017) list promising candidates, such as intuitive models of numerosity, space, physics, and psychology, each with a minimal set of entities, their properties, and relations connecting them. These models, combined with early abilities for causal explanations, compositionality, and learning to learn, drive learning and generalization in unstructured environments.

Symbolic AI in general, and many cognitive architectures in particular, already embody some of these principles. As a result, a combination of symbolic and deep learning approaches, dubbed neurosymbolic AI, is seen as the next research frontier (Marcus, 2018; Greff et al., 2020; Sarker et al., 2021). This new trend is in many ways reminiscent of the hybrid cognitive architectures, which also take advantage of the benefits of symbolic and subsymbolic representations and learning methods by combining them in one system.

Meanwhile, mainstream deep learning continues to pursue a "scaling is all you need" approach, inspired by numerous empirical demonstrations that solve many generalization issues by simply expanding the volume and diversity of training data and increasing the size of models (Geirhos et al., 2021; Beal et al., 2022; Mayo et al., 2022). Even though larger models trained on more data perform better on benchmarks, it remains unclear how much of the improvement is due to memorization or interpolation within the enormous training corpora, as opposed to true generalization. The next section examines this question in more detail.

Scale

Recent scaling trends focus on increasing both the amounts of training data and the number of parameters in models, as their effects are generally more predictable than those of different architectures (Hestness et al., 2017; Zhao et al., 2021).

The amount of data that modern deep networks require for training already exceeds what an average human can perceive in their lifetime. For example, according to a back-of-the-envelope calculation by Tsotsos and Luo (2021), a person who lives to 90 will see 7,568,640,000 images (assuming 16 waking hours and 4 saccades per second). Although this set of images is large, it is highly redundant; most people spend over 80% of their waking time working or at home. Of the remaining 1,513,728,000 images, only 1,645,120 will come from a different geographic location (assuming two-week annual vacations). Other sources, such as books, TV, social media, and video games, contribute to the diversity of visual experiences, but do not change the fact that much of what we see in our lifetime is the same. For reference, the largest public

image dataset LAION contains links to 5 billion images gathered from the Internet (Schuhmann et al., 2022). While the exact composition of the dataset is unknown, it presumably contains a variety of images from all over the world. However, it is safe to say that it is not representative of the visual experience of an average person. For example, it would take 60 years just to glance through these images given the assumptions above.

For text data, the difference is even more striking. Assuming 4.5 hours of reading per day (White et al., 2010), a reading speed of 300 words per minute (Brysbaert, 2019), and a lifespan of 90 years, a person will read 2,513,025,000 words if they learn to read at age 5. For comparison, one of the largest text datasets, The Pile (Gao et al., 2020), contains over 300 billion tokens or 225 billion words—two orders of magnitude more than any human can hope to read.

In comparison, humans learn more efficiently. Most can learn from just a few examples (Berko, 1958) and sometimes even without seeing any examples at all (Malaviya et al., 2022). This suggests that an entirely different learning strategy is employed. While learning from several examples (referred to as few-, one-, and zero-shot learning) can be replicated in deep learning models, pretraining on a large dataset (via backpropagation) is still needed to build an initial knowledge base. Only then can the model learn from new unseen samples representing other related tasks or domains (Wang et al., 2020).

Model scale is seen as another crucial component to boosting performance and generalization (Rosenfeld, 2021). Scaling of training is effective if models have sufficient capacity (Sun et al., 2017). Scaling trends are well illustrated by the foundation models that include text-to-image generators and large language models (LLMs). These models have relatively simple Transformer-based architectures and are trained on simple tasks, such as joint autoregressive prediction of text and images and next token prediction, respectively. Yet, they can perform a large variety of language and vision tasks that they were not specifically trained to perform (Bommasani et al., 2021).

These phenomena led to hypotheses that models over a certain size exhibit a qualitative jump in their capabilities. According to Wei et al.'s (2022) estimates, LLMs with over 68 billion parameters, start displaying other abilities not explicitly related to the training objective of the next token prediction, such as following instructions, chain-of-thought reasoning, addition, and subtraction.

The existence of such emergent abilities has been contested by Schaeffer et al. (2024), who showed that simply changing the metrics used for evaluation turns the sudden spike in the performance into a smoothly growing function, more consistent with power law scaling laws discussed earlier. Additionally, some performance improvements can be attributed to data leakage or the presence of test data in the training set (Aiyappa et al., 2023).

The implications of these findings are profound. If indeed there are no emergent abilities and no qualitative jumps in performance due to scale, then what we observe is not generalization but rather models improving at interpolating between training samples that represent larger portions of the task domain (Marcus, 2022). However, scaling of training data, models, and computational resources follows a power law (Rosenfeld, 2021). As a result, a tenfold improvement of test error would require several orders of magnitude more data and compute (Thompson, 2021). This trend cannot continue indefinitely for

several reasons:

Brute-force learning is ineffective. The space for all visual and textual data is very large and has a long-tail distribution (Udandarao et al., 2024), thus sampling rare samples is infeasible.

The influx of new data is limited. At this rate (and assuming no other changes), new training data will run out in the next ten to twenty years with no guarantee that the foundation models will reach the desired performance level (Villalobos et al., 2024).

The amount of synthetic training data is increasing. A large portion of training data for foundation models is scraped from the Internet. However, model-generated content is being continuously added to the mix. Recent findings show that generative models trained on their own output learn a degenerate representation of data, which can significantly degrade their performance. This phenomenon has already received several names: model collapse (Shumailov et al., 2023), model autophagy disorder (Alemohammad et al., 2024), and Habsburg AI (as a reference to multigenerational inbreeding in the Spanish dynasty that eventually caused its downfall) (Sadowski, 2023).

Cost

The brute-force approach to learning by scaling data, models, and computational resources comes at a cost. Training the models with tens of billions of parameters is estimated to cost millions of dollars for a single run (Sharir et al., 2020) requiring multiple GPU years (Hellendoorn and Sawant, 2021). This excludes "hidden costs" that are often not reported, such as fine-tuning the network architecture and selecting hyperparameters, which may require hundreds of additional runs, albeit usually on a smaller scale (Hellendoorn et al., 2019). The consequences of using more computing power and data to train larger models are not only significant energy consumption (Strubell et al., 2019) and environmental harm (Bender et al., 2021), but also pricing academia out of research (Hellendoorn and Sawant, 2021).

Estimates of brain energy consumption and number of operations performed per second vary by orders of magnitude (Sandberg and Bostrom, 2008, Appendix A), but all indicate that humans are much more energy-efficient than current deep learning models. Conservatively, the human brain uses 10–20 W of power (Merkle, 1989), has $8.6 \cdot 10^{10}$ neurons (Azevedo et al., 2009), and performs 10^{13}–10^{16} operations per second[1] (Merkle, 1989; Furber and Temple, 2007; Grace and Christiano, 2015).

Moravec (1998) made a prediction that by 2020, computer hardware would match the human brain. By some metrics, this has come true. In 2024, top-of-the-line NVIDIA A100 GPU marketed as an AI supercomputer chip, used 400 W at peak, had $54 \cdot 10^9$ transistors, and was capable of computing over 10^{13} FLOPS.[2] By this metric, it may appear that technology is approaching the human brain in some aspects. However, this effectively captures only the

[1] Because the brain is not equivalent to a computer, its computational rate is measured as the number of synapses traversed between the neurons per second instead of floating operations (FLOPS) or millions of instructions (MIPS) commonly used in hardware benchmarks.

[2] https://www.nvidia.com/en-us/data-center/a100/

volume of the computation, but not its nature, thus parity between the GPUs and the human brain is not definitive.

11.2.2 Deep neural networks as models of the brain

The goal of neuroscience is to explain the emergence of observed behavior from neural activity in humans and animals performing a broad range of tasks in their usual environments. The recent success of deep neural networks on machine learning benchmarks has led to an ongoing debate about their utility as models of the human brain. There is already a sizable literature on this subject, and below we will summarize the key points brought up by both proponents and critics of using deep learning in neuroscience.

Anticipated opportunities

Numerous suggestions to integrate deep learning into neuroscience have been made in recent years (Dupoux, 2018; Kietzmann et al., 2018; Cichy and Kaiser, 2019; Richards et al., 2019; Storrs and Kriegeskorte, 2020; Saxe et al., 2021; Doerig et al., 2023). Most single out two properties of neural networks to present them as desirable candidates for models of the brain: multiple levels of abstraction (tying outputs to single-unit and multi-unit dynamics) and performance on real-world tasks. Given these characteristics, they propose the following paths for further investigation.

Testing hypotheses. One of the intended uses of neural networks is for testing and refining existing hypotheses (Cichy and Kaiser, 2019; Saxe et al., 2021; Doerig et al., 2023). This work is still in its infancy, but a number of potential directions to explore are proposed, such as using the neural networks to tackle long-standing questions regarding the role of innate vs. acquired knowledge, importance of supervised vs. unsupervised learning mechanisms, temporal dynamics of learning, the process of forming abstract concepts from perceptual information, and more. To do so, one would compare performance and internal representations of deep networks to human behavioral and neural data on a large variety of perceptual, linguistic, and motor tasks.

Generating explanations. Because testing hypotheses can tell us only what aspects of models correlate with human data, it is suggested that more insight into processes that led to the observed outputs can be obtained via ablation studies and analyses of internal representations. For example, a complex model can be gradually reduced to a simpler and more interpretable form, as Tanaka et al. (2019) did with deep models of the retina. In addition, a variety of methods have been developed for examining computational units in the network, such as visualizing activity (Zeiler and Fergus, 2014; Olah et al., 2017), evaluating their statistical properties (Shu and Zhu, 2019), and establishing their semantics (Zhou et al., 2018). Some see potential in applying similar methods to analyzing biological data (Freeman, 2015).

Generating new hypotheses. It is speculated that analysis of deep learning performance on tasks may give rise to new testable predictions about the human brain. One study claims that responses of AlexNet trained on object recognition helped identify a space of object representations in the primate inferotemporal (IT) cortex (Marblestone et al., 2016; Bao et al., 2020).

Decoding neural signals. Another prospective application is prosthetic limb control and brain-computer interfaces that rely on neural control signals obtained through non-invasive means, such as electromyograms (EMG) for skeletal muscle activation or electroencephalograms (EEGs) for brain activity. Translating these signals into motor commands for physical or virtual devices is a non-trivial task. With the wider availability of large EMG and EEG datasets, deep learning models have began to outperform classical machine learning techniques and are being considered as a potential replacement (Rashid et al., 2020; Jiang et al., 2023).

Existing challenges

As a relatively new undertaking, deep learning methods have yet to deliver fully on their promised potential for advancing neuroscience. Given the nature of deep networks, there are concerns regarding their ability to do so (Forbus et al., 2017; Marcus, 2018; Bowers et al., 2023). The most obvious issue is the lack of biological plausibility, which was discussed earlier in this section. A natural question arises: given that deep learning models are highly abstracted representations of the brain, why should one expect them to reveal anything meaningful about human biology? Even considering that some properties of the biological neurons are captured in deep neural networks, as of now their explanatory and predictive power remains limited.

Evaluation fairness. The conditions in which human performance is measured are not always adequately represented when evaluating models. Canaan et al. (2019) identified the following dimensions that affect fairness of human-machine comparisons: *input space* (e.g. access to information not available to humans, lack of foveation in most vision algorithms), *output space* (e.g. machines do not physically press buttons and may have advantage in reaction speed), *experience* (e.g. an algorithm may be trained on more data than humans), *knowledge* (e.g. an algorithm may receive expertly compiled knowledge about all aspects of the problem that a human subject may not have access to), *compute* (e.g. machines require more resources than humans), and *psychological* (e.g. unlike machines, humans are affected by their physical and emotional state).

Although these dimensions were identified primarily for the game-playing setting, it is easy to see that many apply to other domains. However, regardless of the application, the information on how comparisons were conducted is often incomplete, providing an inaccurate context in which the results are interpreted (Martínez-Plumed et al., 2018).

Explainability issues. One of the goals of cognitive science is to explain how the human brain operates. At the very least, an adequate computational model should match human outputs for the same stimuli. However, there is already sufficient evidence that many deep learning models for visual tasks behave differently from humans: they do not generalize to even modest amounts of noise (Geirhos et al., 2018), are susceptible to adversarial examples that do not affect humans (Szegedy et al., 2014), and respond differently to psychophysical stimuli (Bowers et al., 2023).

Models that are concerned not only with the input-output transformation, but also with how it is accomplished in the brain are more desirable. The utility of deep learning as a tool for this type of explanation hinges on the

assumption that models that reach human performance use similar neural mechanisms. However, recent empirical work casts doubt on the feasibility of this line of research by showing that high task performance does not necessarily lead to development of brain-like representations.

Recently, Schrimpf et al. (2018) tested Yamins et al.'s (2014) earlier claim that high image classification accuracy correlates with the ability to predict neural response data. According to the new analysis, not all models that achieve $\geq 70\%$ top-1 accuracy on ImageNet are the best predictors of neural or behavioral data and vice versa. Moreover, architectures inspired by the ventral pathway (e.g. AlexNet[3], VGG; Simonyan and Zisserman, 2014) score better than architectures that are not (e.g. PNASNet; Liu et al., 2018). Schaeffer et al.'s (2022) study presents similar results for the path integration task; again, many models could be trained to perform the task but a mere fraction of them developed brain-like grid cells and only when specific feature encoding was chosen for the readout layer. It appears that the internal structures are not necessarily fundamental to the task, rather they depend on the architecture and input representation. Tuckute et al. (2022) reached a similar conclusion by examining deep learning models trained on a range of auditory tasks and tested on neural responses in auditory cortex. Analysis revealed that choice of the task critically affected how well models predicted brain activity.

These examples demonstrate that it may be possible to model known phenomena within the neural networks but their ability to explain observed behaviors and make new predictions is still limited. If neural structures do not emerge automatically from learning to perform the task, then other factors, such as architecture, training data, training regimen, and input representation must be considered. Because the space of possibilities is very large, this path toward finding mechanistic explanations may not be a viable option.

Interpretability issues. The opaqueness of deep neural networks presents another challenge. It may seem contradictory because the computational units of neural networks are much simpler and far less diverse than biological neurons. Rather, it is the scale of the models, some of which comprise billions of units, that makes them incomprehensible. Now, instead of one unknown system (brain), there are two (brain and neural network).

The spread of deep models in critical domains, such as autonomous driving and medical diagnosis, intensified efforts to "open the black box" and explain its performance in a human-understandable way. A recent comprehensive survey by Schwalbe and Finzel (2023) lists dozens of such methods, most of which lack reliability and offer at best partial explanations. Among them, various methods that attribute features of the input to the output of the network are some of the most popular, however, they do not provide causal understanding of unit activations (Kindermans et al., 2019; Leavitt and Morcos, 2020; Zimmermann et al., 2021). It is also possible to examine a neural network by extracting the if-then rules describing its behavior. While it worked well for smaller networks (Andrews et al., 1995), finding rules in much larger deep learning networks is computationally expensive, sensitive to initialization, and can produce very large sets of rules (Gilpin et al., 2018; Ras et al., 2018).

[3]AlexNet architecture, along with other approaches, e.g. LeCun and Cortes (1998) and Cireşan et al. (2011), can be traced to the hierarchical structure of Fukushima's (1980) Neocognitron that modeled "simple" and "complex" cells found in a cat's

Explanations on a deeper level, akin to having a discussion with a colleague and reaching a conclusion together, are still out of reach.

11.3 Deep learning in cognitive architectures

Historically, connectionist or subsymbolic representations were part of many cognitive architectures (see Chapter 3). However, deep learning, being recent relative to the timelines of most cognitive architectures, has not had as much impact yet. The integration of neural networks into cognitive architectures has followed two paths: 1) modular integration—using networks as black boxes for solving practical issues and 2) neurosymbolic integration—development of cognitively and biologically plausible neural representations and learning approaches. Both will be discussed in more detail below.

11.3.1 Modular integration

The simplest way of integrating a neural network into an existing cognitive architecture is in a form of an encapsulated module. Before pretrained models for a variety of tasks have become widely available, neural networks were custom designed and trained on the data collected for the specific application. Whether custom-made or off-the-shelf, the ANN is usually frozen after training and turned into a module that receives input and produces output at runtime. The main purpose for integrating such neural networks is solving practical problems that are less amenable to symbolic methods, such as perception and motor control.

Perception. Visual processing due to its complexity and non-linearity remained particularly resistant to heuristic methods and symbolic manipulation. As a result, many cognitive architectures that operated in realistic environments resorted to using ANNs for processing sensor data. In the 1990s, simple MLPs were being used for various tasks. For example, a person-following robot based on the 3T architecture performed face cognition via a simple three-layer network (Wong et al., 1995). An assistive robot ISAC likewise relied on a small neural network to recognize objects and people in the environment (Kawamura et al., 1995). Recent cognitive architectures use CNNs for similar purposes. For example, the visual hierarchy of the STAR-RT model for playing video games was implemented as a custom four-layer CNN. The network was trained to detect and recognize multiple classes of game objects, such as characters and obstacles (Kotseruba, 2016).

Pretrained models are even easier to integrate, as they require little to no fine-tuning to perform basic vision tasks. One example is the STAR-FC model of fixation control that relied on a pretrained CNN-based saliency detector to identify candidate areas to fixate (Wloka et al., 2018). A LIDA model of spatial memory used several pretrained CNNs for road detection, semantic segmentation, and object recognition to identify landmarks in a simulated traffic environment (Madl et al., 2016). Similarly, an assistive robot controlled by the CORTEX architecture located common office items via an off-the-shelf CNN-based object detector (Mendoza et al., 2018).

Motor control. Many cognitive architectures have been embodied on complex robotic platforms with many degrees of freedom. In such cases, analytical

control solutions may be too complex or brittle, and learning methods are more advantageous. For example, a neural network for controlling articulated robotic arms and a multilegged robots is advantageous because many degrees of freedom and coordination of several actuators made it difficult to construct a sufficiently adaptive and quick closed-form solution (Bekey, 1996). Similar considerations guided the design of motor control for the robotic arm and visual system of ISAC, a humanoid robot based on the IMA architecture. Here, a neural network was used to map the joint angles of the robotic arms to input voltages of the valves for the pneumatic actuators. The use of the neural network helped overcome issues with hysteresis and non-linearity of motion (Ulutas et al., 2008). Additionally, ISAC's gaze was controlled by a neural network, which handled the non-linear motion and coordination of two cameras without the need for accurate camera calibration (Peng et al., 2003).

Memory and decision-making. Although perceptual and motor tasks are a more natural application of neural networks, there have been attempts to use them for memory and decision-making. For example, working memory in the IMA is implemented as a neural network trained to retain chunks of information by estimating their future reward. The temporal difference (TD) error is calculated from the previous time step and compared to the estimated value to update the weights of the network (Kawamura et al., 2008). In IDA, a simple feed-forward neural network augments symbol manipulation during decision-making. A learning approach is chosen to ensure that the output conforms to a set of predefined soft constraints (Kelemen et al., 2003).

Modular integration of neural networks remained a niche solution. Among the projects we reviewed, few were committed to purely symbolic methods while others were explicitly agnostic to underlying representations (e.g. ARCADIA, FORR, and Polyscheme). This suggests that, in principle, many more cognitive architectures could have incorporated ANNs. Thus, a more likely reason is that fewer sensors and low complexity of most perceptual and motor problems could be solved with heuristics—until now. As cognitive architectures are interfaced with more and better sensors and tested in more realistic scenarios, there may be a need for more sophisticated techniques, including neural networks. Large language models (LLMs), in particular, are seen as a potentially useful tools (Joshi and Ustun, 2023; Knowles et al., 2023; Williams et al., 2024).

Despite this potential, there are disadvantages to using neural networks that limit their utility for cognitive architectures. These include biological implausibility, lack of transparency, and issues with generalization discussed earlier. In addition, a frozen neural network encapsulated in a module cannot adapt effectively. If any changes are introduced to the cognitive architecture, task, or the environment, the network often needs to be retrained. Lastly, large amounts of data needed for training a model may be difficult to obtain and annotate.

11.3.2 Representational integration

A deeper integration of connectionist representations places them at the core of the architecture rather than isolating them in separate modules. Usually, choice of neuron models, connectivity between them, and learning algorithms

are guided by the requirements of cognitive and biological plausibility. In addition, connectionist representations may be combined with symbolic ones to capture the strengths of both approaches. The resulting models of brain mechanisms are more biologically plausible, interpretable, and not as computationally expensive as mainstream neural networks.

Representations. Only several architectures use biologically plausible neurons: leaky integrate-and-fire (LIF) in the instance of SPA (Stewart and Eliasmith, 2012), adaptive exponential (AdEx) model of pyramidal neurons in Leabra (O'Reilly et al., 2016), Izhikevich model in a humanoid BBD (Chen et al., 2013), and Hodgkin-Huxley neuron in one of the ART instances (Grossberg and Versace, 2008).

MLP-like neurons appear in visual hierarchies of STAR and SASE, but modifications are made to make them more biologically plausible. For example, in STAR models of shape encoding and border ownership, layers, areas, and connections among them are initialized to match data from neural recordings (Mehrani and Tsotsos, 2021; Mehrani and Tsotsos, 2023). In SASE, computational units are organized in a 2D array and connected only to a restricted set of units in the preceding layer, emulating a limited receptive field that grows larger in the top layers (Zhang et al., 2002).

Others are composed of more explicit representations to denote concepts, properties, and events. Semantic pointers in SPA are vector representations of populations of spiking neurons. Operators, such as binding and concatenation, enable concept manipulation (Stewart and Eliasmith, 2012). The nodes in DeSTIN and BECCA represent clusters of input features (Goertzel et al., 2014; Rohrer, 2013). In SHRUTI and LISA, basic units are semantic tokens that represent neuron populations (Shastri, 1999; Hummel and Holyoak, 2003). DUAL uses symbolic frames instead (Kokinov, 2013). In all these systems, subsymbolic elements are represented by the unit activations, connections between units, and strengths of connections. Clarion combines several of these strategies: actions are represented by symbolic rules and MLPs, whereas concepts are encoded via symbolic and Hopfield-type connectionist networks (Sun and Helie, 2013).

Hierarchical structure. Most connectionist and hybrid representations are organized in multilayered graph structures that mimic the structure of the human cortex. At the lower level, representations are derived directly from perceptual inputs and subsequent layers contain increasingly abstract and compressed versions of the original input.

Semantic pointers in SPA retain semantic information about what they were derived from, hence the name (Blouw et al., 2016). DeSTIN and BECCA incrementally build feature representations from various types of perceptual inputs. The nodes in DeSTIN predict the probability of the centroid given in the observation and past history of the observations. The nodes feed inputs to their parents and receive advice from them to condition probability calculations on the context (Goertzel et al., 2014). In BECCA, the nodes compute centroids through repeated application of clustering to inputs and features, resulting in a hierarchical feature structure with an arbitrary number of levels and clusters within each level (Rohrer, 2013). In a symbolic hierarchy of LISA, semantic units at the bottom correspond to tokens ('male', 'female', 'has-emotion'), which are gradually combined into objects ('Bill', 'Mary', 'lover', 'beloved'),

predicates ('Bill+lover' and 'Mary+beloved'), and relations ('Bill loves Mary') via connections with weights. When the system is given input, the weights of the connections in the network are updated, formed, or destroyed, giving rise to new relations between concepts (Hummel and Holyoak, 1997).

Learning methods. Backpropagation is widely regarded as the most effective learning method, but its use in cognitive architectures is limited due to biological implausibility, as discussed earlier in Section 11.2. Among the few exceptions are SHRUTI and Leabra. SHRUTI incorporates supervised learning with backpropagation (Shastri, 1999), although it can also accommodate other types of learning. Leabra uses a more biologically feasible form of backpropagation derived from Hinton and McClelland's (1987) recirculation algorithm (O'Reilly, 1996).

ART takes a different path toward biologically plausible learning with an Adaptive Resonance approach. It resolves a number of issues with backpropagation, notably dependency on supervision, mostly off-line learning, and catastrophic forgetting. Most ART networks combine a feedforward bottom-up pass with two types of feedback interactions: contrast normalization and top-down expectations that focus attention on critical feature patterns and suppress irrelevant features and noise (Grossberg, 2020).

Another common biologically feasible alternative to backpropagation is Hebbian associative learning found in the architectures that use associative networks, e.g. SHRUTI (Wendelken and Shastri, 2003), MBCA (Schneider, 2019), Clarion (Sun and Helie, 2013), HCA (Lesser et al., 2008), BECCA (Rohrer, 2013).

Reinforcement learning is yet another possibility explored in many architectures, often in conjunction with other learning methods. For example, BECCA, DeSTIN, and SASE combine unsupervised clustering for feature learning and reinforcement learning for action selection. While DeSTIN uses a standard actor-critic reinforcement learning method (Goertzel et al., 2014), the other two introduce modifications. BECCA introduces a variant of TD learning called S-learning, which is superficially similar to Q-learning (Rohrer, 2007a). SASE combines elements of unsupervised, supervised, and reinforcement learning. Instead of backpropagation, SASE uses a modified principal component analysis (PCA) to incrementally learn filters for visual processing (Zhang et al., 2005). For procedural knowledge, SASE extends a typical Q-learning RL framework by allowing a human instructor to impose internal state or state of the effectors of the agent through natural language commands (Weng, 2007). Clarion combines symbolic rule extraction with Q-learning to learn and refine its procedural knowledge (Sun et al., 2007).

11.4 Can deep learning result in a cognitive architecture?

Today many see deep learning as the most promising path toward human-level intelligence and beyond. These hopes are supported by the "superhuman" performance of deep learning algorithms on some tasks, such as object detection (He et al., 2015), playing various games, such as chess (Silver et al., 2016), classic Atari computer games (Mnih et al., 2015), and poker (Brown and Sandholm, 2019), car racing (Wurman et al., 2022), and more.

Although some of these performance claims have since been disproved[4] or at least weakened,[5] deep learning is still perceived by the research community as the most successful approximation of human intelligence (see comments to Bowers et al.'s (2023) review). We argue that while the current approach is effective in practice, it is unlikely that it will result in anything resembling the architecture of the human brain.

Deep learning research is a mostly bottom-up process that starts with restating a problem in the form suitable for learning and searching for the best technique for solving it. As a result, the organization of deep learning systems depends on the task rather than the cognitive function it requires. Networks include only those components necessary for the task, and designs are different for each. As there is no theoretical foundation, most of the work is empirical. Consequently, justifications of the design and implementation choices are often omitted or obtained post-hoc. Even if this process eventually converges to a parsimonious set of computational mechanisms that enable human-level performance, it is not guaranteed to be human-like. Already, the most performant deep learning architectures are also the least similar to human brain and behavior, as discussed earlier in Section 11.2.

In contrast, cognitive architectures aim to discover and explain the essence of human cognition by constructing artifacts that mimic it. Thus, some level of cognitive or even biological plausibility is present by design. The architectures also embody concrete theoretical proposals about the human mind and brain; while empirical methods may be used locally to fill gaps in theory or to tune parameters, the overall process is largely top-down—starting with specific premises and constraints that guide design and implementation. As a result, different cognitive architectures share a lot of high-level structure (see Chapter 8). For example, in all projects that we considered, working memory plays a central function due to its importance for human cognition.

Lastly, building and maintaining cognitive architectures is a long and arduous process that demands multidisciplinary expertise; the majority of projects take years of work before producing concrete results. Thus, continuity and consistency of vision are essential. Not surprisingly, long-standing projects in our selection of architectures are led by the same core teams of researchers, some for many decades. In comparison, there are few stable projects in mainstream deep learning that are continuously developed by the same group. Instead, anyone can potentially contribute thanks to the availability of frameworks, pretrained models, and computational resources that have considerably lowered the barriers to entry in recent years. However, the ease of experimentation, publication pressure in academia, and pursuit of short-term business interests

[4]Toromanoff et al. (2019) argue that the metrics used to compare performance of DQN (Mnih et al., 2015) on Atari games underrepresented human performance, therefore the claims of superhuman performance are misleading.

[5]"Superhuman" models regardless of learning method, task, and training data can make silly mistakes that most humans will not, which is sometimes referred to as adversarial examples. For instance, image classifiers trained via supervised learning on large image sets tend to overgeneralize concepts (e.g. confuse bicycles and tricycles), rely too much on texture or shape, and misclassify associated objects (e.g. hummingbirds with hummingbird feeders and snow with shovels) (Hendrycks et al., 2021). Likewise, seemingly unbeatable reinforcement-learning-based Go-playing AI systems are vulnerable to certain classes of strategies that allow non-experts to consistently win against them (Wang et al., 2022).

in industry have also led to an explosion of useful but ultimately incremental work, whereas many fundamental issues remain unaddressed.

11.5 Summary

- Conceptually, deep learning traces its roots from connectionism in the early days of AI. But only recently, advances in computing hardware, improvements in learning algorithms, and the availability of large datasets have enabled successful practical applications of deep learning in many domains. As a result, it is now the prevalent paradigm in AI, pursued by both academia and industry.
- Despite its dominance in AI, deep learning has no significant presence in the field of cognitive architectures. There are a few architectures that incorporate shallow neural networks as perceptual and motor modules and several more that focus on biologically inspired connectionism.
- The relationship between neuroscience, cognitive science, and deep learning is skewed. Mainstream deep learning pursues scaling at all costs and largely ignores biological plausibility, except when it contributes to practical success. On the other hand, neuroscience and cognitive science mainly contribute to identifying the weaknesses and limitations that help improve designs of neural networks and benchmarks.
- Although deep learning is seen by some as a way to bridge neural computation and behaviors, the usefulness of neural networks as models of the human brain and cognition is still unproven.
- Given the inherent limitations of deep learning and the bottom-up approach to modeling cognition, it is unlikely that the current research paradigm will lead to the emergence of a human-like cognitive architecture.

12 Challenges of the Past and Opportunities Ahead

The main purpose of this book is to catalog and summarize the past and present efforts toward modeling the human mind and brain in cognitive architectures and to describe their successes and limitations. We looked at the goals and roots of cognitive architectures, identified common approaches for modeling important elements of human cognition, established practical abilities of cognitive architectures, discussed evaluation procedures, and, lastly, the place of cognitive architectures in the era of deep learning.

Throughout the book, we have refrained from injecting judgments or personal opinions, and, instead, tried to rely on the publications we reviewed to identify important problems. We conclude the book with this chapter, where in addition to the views of the community at large, we provide our thoughts and commentary on the issues that the field is facing today and suggest directions for future research.

Section 12.1 describes current challenges facing cognitive architectures as a field. Here, we consider the range and fidelity of cognitive abilities as well as issues with evaluation and reproducibility.
Section 12.2 turns lessons learned from the past forty years of cognitive architectures into advice for robust research and development practices and open problems.

12.1 Current limitations

When it comes to challenges and limitations of cognitive architectures, there is no better source of insights than people who specialize in designing and developing them. To gather perspectives of the wider community, we identified papers in our corpus that explicitly discussed the limitations of the systems and issues that were encountered during the development and evaluation. We then categorized the issues into groups by cognitive abilities to better match them to the structure of the discussion as follows: perception, memory, learning, and reasoning, as well as challenges associated with system integration, evaluation, and performance. The plot in Figure 12.1 visualizes the relative frequency of mentions for each category.

The observed distribution of concerns, where perception and evaluation are most prominent, is consistent with our own analysis in the previous chapters. The weak state of perceptual capability, ranging from its absence, its simplification via restrictive assumptions, and poor noise tolerance to the lack of attentional mechanisms and active perception are reported across the majority of cognitive architectures, both historic and modern ones. With regard to evaluation, a large portion of the papers point

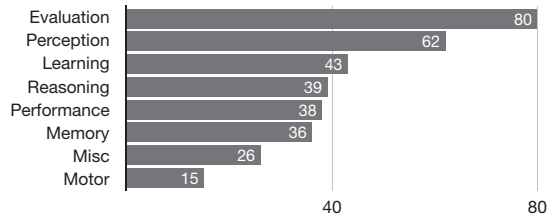

Fig. 12.1 Issues corresponding to the core cognitive abilities discussed in the Part II of the book as well as evaluation and performance. The information is extracted from 329 papers (12% of the total 2,800 papers in our corpus) which explicitly discussed the issues and limitations of the respective approaches.

data as the main concern, while others advocate for testing architectures on more varied, challenging, and realistic tasks over longer periods of time. There are, however, only a few mentions of the lack of quantitative comparative evaluations as an issue to be addressed, again agreeing with the conclusions in Chapter 10.

The limited learning ability and consequent overreliance on hard-coded knowledge are widely acknowledged weaknesses in most cognitive architectures, also in line with our observations. Somewhat surprisingly, issues related to reasoning were emphasized, despite these abilities being a strong suit of many cognitive architectures. Here, inability to deal with failures, interruptions, and inconsistent knowledge, along with the related problem of limited meta-cognition, received the most attention.

Performance-wise, the number one concern is computational complexity that causes slow execution time and the difficulties with scaling to larger knowledge bases. While responsiveness is particularly critical for robotic architectures, slow running simulations are also unfavorable for general cognitive architecture research because they restrict the scope of experimentation and complicate comparisons with human data.

Issues categorized as miscellaneous in Figure 12.1 form a connected cluster. The key problems are the impaired social and communication abilities of most cognitive architectures, which in turn are related to a lack of deep system integration between sensing, motor actions, and reasoning.

The rest of this section will go into more detail of the key challenges in each of the research areas. We will also discuss problems of reproducibility and replicability that have received little attention in the literature so far.

12.1.1 Range and realism of cognitive abilities

The dozens of cognitive abilities and phenomena related to perception, memory, learning, decision-making, and reasoning that have been covered in this book are by no means representative of the full breadth of human competence. In fact, none of the examined cognitive architectures support in theory or demonstrate in practice even a basic set of human abilities, thus falling short of representing the nearly 3,000 distinct human cognitive abilities listed in Carroll's (1993) comprehensive survey. Only a small fraction of architectures explores creative problem-solving, communication, social interaction, natural language understanding, and complex motor control. Even for some of the most

basic abilities, cognitive architectures do not yet provide complete mechanistic explanations or realistic behavioral outputs.

Perception

The treatment of perception in cognitive architectures is at odds with developments in both artificial intelligence (AI) and cognitive science. Some of the largest growing conferences and fastest growing literature are in computer vision. The best understood part of the human brain is widely believed to be the visual cortex. Yet, in cognitive architectures, perception is arguably the most understudied of the core cognitive abilities.

Historically, most cognitive architectures primarily focused on reasoning and planning in simplified domains, often overlooking the interaction between perception, memory, cognition, and attention mechanisms that help deal with the vast complexity of sensory data. As a result, the treatment of perception in most cognitive architectures remains superficial, falling short both in terms of capturing behavioral phenomena and replicating neural mechanisms of human perception.

Although vision is by far the most common sensory modality, it is simulated in more than half of the cognitive architectures we considered. A common approach is to use highly preprocessed sensory data, such as the scene already parsed into objects with all their properties and spatial relations. Needless to say, such simplifications not only result in systems that struggle with more realistic data, but may also affect cognitive abilities that depend on perception. For example, working memory designed for pristine well-defined data will likely lack mechanisms for rectifying errors and dealing with noise.

Even in robotic systems that must implement perception with noisy sensors to satisfy strict performance requirements, the solutions are often suboptimal. Many involve highly specialized pipelines with unrealistic assumptions to simplify processing as much as possible. Consequently, these systems can only be deployed in simulated or highly controlled real environments. Very few cognitively plausible vision systems attempt to counteract the issues of noise, complexity, and volume of sensory data via attention and active perception. Usually, attention is more often a side effect of other design decisions rather than an intentional choice. Overall, visual perception in cognitive architectures lags significantly behind its biological counterpart in terms of robustness and generalizability.

The processing of sensory modalities other than vision, such as audition, proprioception, and touch, is often delegated to off-the-shelf software. While these solutions can save labor and offer practical benefits in the short term, they have a significant downside. When it comes to interaction with the architecture's components, external software is by its nature difficult to fully integrate with the rest of the system. For instance, internal representations within external modules are often not accessible, especially if proprietary binaries are being used. Conversely, representations from external modules that are not compatible with the rest of the architecture may require workarounds or even prompt additional modifications to the architecture itself.

Memory

Memory due to its central role in cognition is the only component that is found in all cognitive architectures. Because of the significant differences across

cognitive architectures in terms of their theoretical backgrounds, structure, representations, and processing, memory implementations are very diverse. The level of theoretical and technical specification also stands out compared to other abilities. Nevertheless, memory is far from being a solved problem.

The vast majority of memory systems we reviewed in this book are purely functional and primarily serve decision-making without much regard for cognitive or biological constraints. Simplifications and ad hoc solutions are often introduced, either motivated by performance considerations or because of gaps in knowledge about human memory. For example, many memory models are effectively unlimited in size, possess perfect recall, and forget nothing.

Furthermore, computational models of memory are rarely assessed on their own or compared to how human memory operates. However, even the limited available evidence from our analysis shows that not many memory models have a potential of reaching the scale and efficiency of their biological counterparts, and even fewer can effectively explain human memory mechanisms.

Another issue is the notable absence of episodic memory. Despite being discovered decades ago and widely recognized for its importance in learning, communication, and meta-cognition, episodic memory remains largely ignored in cognitive architectures; less than one-third of systems include it, usually in a rudimentary form. The most common approach to episodic memory is building a database of time-stamped snapshots with simple consolidation mechanisms to remove duplicates and simplify search.

Lastly, although the majority of the systems segregate memory by type, duration, or both, the links between them are not sufficiently explored. These include interactions between short- and long-term memories and episodic and general knowledge, mutual effects of declarative and procedural knowledge on one another, and dealing with inconsistent or erroneous memory items.

Learning

Many types of learning have been investigated and implemented across cognitive architectures, but there is still more to do. One-third of cognitive architectures do not implement learning. Instead, memory contents are hard-coded, which is very labor-intensive and requires considerable expertise in the given task. Thus, when evaluating the cognitive architecture that acts on such knowledge, it becomes difficult to discern the relative contribution of the implemented mechanisms and human programmer's ingenuity to the observed performance.

The remaining two-thirds of the architectures with learning do not come close to the full range of human learning ability. For instance, some are limited to learning only procedural or semantic knowledge, i.e. they can learn new tasks but not new factual knowledge or vice versa. Others learn different types of knowledge but are limited in the types of learning mechanisms, such as only learning through trial and error or by instruction, but not both, as humans do.

Learning itself is often not efficient. Supervised and reinforcement learning scenarios often require thousands upon thousands of trials to master a simple movement or a set of facts. Humans, on the other hand, improve quicker, and in some cases can learn from a single observation or a brief verbal description.

Once something is learned, it often remains frozen in memory. This is particularly true of perception, which in many cognitive architectures relies on

prelearned models for detecting objects and parsing scenes. Any kind of long-term continuous learning remains a challenge, partly due to short durations of most tasks and partly due to limitations of both connectionist and symbolic approaches. For instance, neural networks struggle to balance plasticity with catastrophic forgetting, whereas symbolic systems suffer from brittleness and utility problem as they accumulate larger volumes of knowledge.

Reasoning

Reasoning is another core ability that is well-represented across all cognitive architectures. Despite being a major focus of efforts, robust reasoning and decision-making with incomplete information and under resource constraints remain a major challenge.

Perhaps, the most long-standing issue, reported in the papers from the 1980s as well as more recent work, is dealing with various failures, interruptions, and inconsistencies in the input or knowledge base. Such situations happen during everyday tasks and in safety-critical domains due to distractions, sudden changes in external circumstances, or new internal goals.

Typical solutions focus more on reducing the chance of such incidents by controlling the environment and hard-coding the knowledge to cover all possibilities and externalities, but it is clear that the true solution lies elsewhere. Many agree that whatever form the working solution will take, some form of meta-cognitive ability should be part of it. In order to return to the previous task after being interrupted or assess how to deal with the failure, a system should be able to monitor and analyze its own behavior. However, meta-cognition itself is not widely supported across cognitive architectures and is often too rudimentary to make a difference.

Behavior modulation or internal goal direction can also contribute to more robust, autonomous, and human-like behavior. The effects of personality traits, drives, moods, and emotions on other aspects of cognition, such as memory and decision-making, are well studied in psychology but are featured in only a select few cognitive architectures. Although potentially any cognitive architectures can be "retrofitted" with these abilities, not many examples exist. Even rarer are cognitive architectures designed with behavior modulation in mind. Despite supporting behavior modulation natively, these implementations are quite basic and not comparable to the breadth of emotion and traits displayed by humans.

Social abilities

Displaying and recognizing emotions is necessary for social abilities and effective human-machine communication. Other prerequisites for social competence are robust perception, natural language processing, effective reasoning, and quick response. Not surprisingly, given the issues described above, it is generally acknowledged that cognitive architectures universally struggle with scenarios that involve other robots or humans.

For example, verbal communication is the most common mode of interaction between artificial agents and humans. Many issues are yet to be resolved, as the current approaches do not possess a sufficiently large knowledge base for generating dialogs and generally lack robustness. Few, if any, architectures are capable of detecting the emotional state and intentions of the interlocutor or giving personalized responses. Non-verbal aspects of communication, such

as performing and detecting gestures and facial expressions, natural turn-taking, etc., are investigated by only a handful of architectures no longer in development. As a result, most demonstrated interactions are heavily scripted and often serve as voice control interfaces for supplying instructions instead of enabling real collaboration between the human and the machine.

System integration and multitasking

Even though integration of multiple elements of cognition is one of the goals of cognitive architectures, it has not been fully achieved, as often admitted by researchers themselves. A particular area that needs improvement is the tighter integration of perception and motor abilities with the rest of the cognitive operations. This would not only make architectures more human-like but could also help address the robustness and flexibility issues currently plaguing the field.

One aspect not emphasized in the literature is that most of the tasks and skills attributed to cognitive architectures are implemented as separate instances. As a result, there is no chance of reusing the capabilities or accumulating knowledge from one task to another because every instance has its own design, knowledge base, skill set, and often a specific set of parameters. Furthermore, compatibility of processes across these instances cannot be verified. Although there are attempts at building unified models for performing multiple tasks (e.g. Spaun based on the SPA architecture), most cognitive architectures are a collection of individual models built on top of the core functionality rather than a single integrated system capable of performing multiple tasks.

Additionally, as the underlying core evolves, those instances created with the previous versions may no longer be compatible with the new version and vice versa. This lack of backward compatibility is an issue for long-running projects and adds to the challenge of building and maintaining a coherent system.

12.1.2 Evaluation

By far the most pressing and long-standing issue in the field is the thorough experimental testing of cognitive architectures, as discussed in Part III. For instance, many cognitive architectures (including those that aim to achieve AGI) are not sufficiently validated against human data or compared to other approaches. Additionally, they are rarely tested on challenging benchmarks and realistic environments, and only a few have demonstrated utility in real-world applications.

Validation on human data

Replicating human data from psychological experiments is a natural way of assessing intelligent systems that aim to mimic human performance. It is not the most common evaluation method, but it has been applied to a number of cognitive architectures. Often the reported fit to human data is limited, i.e. the model can adequately represent only some aspects of human performance. Even though the cause of mismatch is not always identified, there are common issues with conducting such studies. Besides small numbers of participants, which limit the statistical power and effect size, the inputs and outputs of the

models rarely match those of humans. Since perception is generally the weak point for most cognitive architectures, the input is highly simplified. In many cases, the knowledge and strategies for performing the task are hard-coded as well. In addition, postprocessing of the output is often required to match it with the temporal resolution or other properties of the human data.

As conclusions are often supported by a cursory statistical analysis, there is no guarantee that a given architecture possesses the same cognitive ability as the human subject performing the task. Considerations that may lead to different interpretations are often ignored. These include model complexity, generalization, and falsifiability. Besides, restricting evaluations to psychological experiments tells us little about the actual abilities of the system in the real world due to the abstract nature of many psychological tasks and inherent limitations of the laboratory setting.

Lastly, there is virtually no overlap in terms of the tasks or specific studies used for evaluation. Therefore, it is not possible to compare performance or approaches to modeling across architectures.

Quantitative and comparative evaluation

In our view, the lack of quantitative and comparative evaluations limits the development of the field. The absence of rigorous evaluation methods makes it challenging to identify the strengths and weaknesses of different approaches, compare their performance, and build upon previous work. Instead, most cognitive architectures are tested on disjointed and sparse sets of experiments, from which it is difficult to infer their abilities and representational power.

Surveys, such as this book and others we mentioned earlier in Section 10.2, are a first step toward mapping the landscape of approaches to modeling human intelligence, but they can only identify broad areas in need of further exploration. For a comprehensive and meaningful analysis of what has worked and what has not, one needs to go beyond the qualitative information that can be extracted from descriptions and demos. In order to faithfully represent the extent of the cognitive ability in practice, quantitative comparisons are also necessary.

For an example of what comparative evaluation can provide, we can look at the recent developments in AI. Up until the early 2000s, the situation was the same as in the field of cognitive architectures now. Much of the work has been evaluated on isolated cases individually or against simple baselines. The situation has changed dramatically in recent years as common benchmarks have began to shape the field by focusing activity around specific problems (Martínez-Plumed et al., 2021). Improved performance on benchmarks has become the main metric for quantifying progress toward solving hard problems in many areas of AI.[1]

It should be noted that treating benchmarks as the most important metric of a scientific contribution has its negatives. Dehghani et al. (2021) liken benchmark-driven research to a lottery, meaning that success of computational approaches is determined by many factors not necessarily related to algorithmic superiority, especially in the absense of rigorous statistical significance

[1] For example, Stanford University has been using benchmarks to compose annual AI Index reports on advancements in the field, see

tests. Another consequence of benchmarks is biasing the field toward engineering solutions and away from fundamental problems (Su and Crandall, 2021).

In sum, while comparative quantitative evaluation is not without issues, benchmarking is a powerful tool when used properly and interpreted with caution. For various reasons discussed in Chapter 10, we do not yet observe a shift in the way cognitive architectures are evaluated. We see this as an opportunity to apply lessons learned in the computer science community to improve quantitative methodologies for testing and comparing cognitive architectures.

Running time and scalability

Evaluations that focus on the efficiency and scalability of cognitive architectures are even rarer, thus there is very limited quantitative information about space and time complexity of cognitive architectures we reviewed.

The problem of computational efficiency is taken more seriously in the architectures that target robotic and interactive applications where responsiveness matters. But overall, cognitive architectures are not designed with performance considerations in mind.

Regarding scalability, it has been shown that applying existing implementations to larger amounts of data is not possible without optimizations to data processing and storage that sometimes result in a complete rewrite. Without such modifications, the majority of cognitive architectures have difficulty maintaining a large knowledge base and operating in complex environments. However, there are still not many works addressing this problem, even though demonstration of advanced cognitive abilities and real-world problems clearly demand further improvements.

12.1.3 Reproducibility and replicability

Reproducibility of results is one of the fundamental requirements for scientific research (Schmidt, 2009). In psychology, reproducibility refers to the ability to obtain the same results by applying the same analysis to the same data (Nosek et al., 2022). An equivalent in machine learning, also referred to as computational reproducibility, involves running the code provided with the study on the same dataset and obtaining the same results (Gibney, 2022). A stronger condition is replicability, which entails independently repeating the same study, reimplementing the algorithm, or recreating a dataset based on the specification.

Both psychology and AI have been dealing with the "reproducibility crisis" in the past decade. In the case of psychology, a large-scale study by the Open Science Collaboration (2015) attempted to replicate a hundred psychological studies but succeeded with only a portion of them. While 97% of the original studies claimed statistically significant results, only 36% of the replications confirmed them. Similar concerns have also been raised for research in machine learning. In one study, out of 255 papers published between 1984 and 2017, only 63.5% were successfully replicated, i.e. for those papers at least 75% of the original claims were confirmed with the independently written code (Raff, 2019).

In both disciplines, a combination of incomplete descriptions, filtering training and test data, incorrect statistical analysis or evaluation of results, HARKing (Hypothesizing After Results are Known), underreporting of and

lack of code and data used for analysis are common causes of replication failures. Additionally, in the machine learning research, data leakage (or contamination of training data with test data) has been identified as a major issue preventing replication and reducing the validity of the results (Kapoor and Narayanan, 2022). Given the variety of these issues and their causes, the proposed remedies are multifaceted as well, ranging from incentives for conducting replication studies to improving transparency of methodologies and accessibility of code and data (Stevens, 2017; Kapoor and Narayanan, 2022).

In the field of cognitive architectures, no large-scale replication studies have been conducted to date, which itself is concerning. Nevertheless, it is very likely that replicability issues exist in this area as well. Next, we consider several problems that impede reproducibility and replicability of results in the field of cognitive architectures.

Code and data availability

Availability of source code and data is seen by many as an important step toward improving replicability in many fields. As a result, journals and conferences across many disciplines have recently begun to encourage or even enforce publication of the code and data (Hardwicke et al., 2018; Pineau et al., 2021; Tedersoo et al., 2021).

Given that the bulk of cognitive architecture papers we considered were published before such measures became widespread, the availability of materials needed for computational reproducibility is sparse. While nearly a half of the cognitive architectures we reviewed maintain a project webpage, only one-third provide code or binaries for the core components. Even fewer release code for individual models and data needed to reproduce the experiments. Finding them is not easy, as less than 2% of the papers have links to the code and studies are rarely associated with the code on the project pages. On a positive note, nearly half of the projects that do share code host it on platforms, such as GitHub and BitBucket, which ease access and encourage collaborative development.

Method specification

Another condition for successful replication of the results is the clarity of technical descriptions (Cooper and Guest, 2014; Raff, 2019). However, the literature on cognitive architectures gives far more importance to the cognitive, psychological, and philosophical aspects of the proposed approaches while underspecifying or omitting technical implementation details. For example, we often encountered publications that described certain capabilities but did not provide enough concrete information on their exact range nor how they were achieved algorithmically. This issue is particularly acute for perception and motor control, although details of other abilities are often not fully specified either.

Even seemingly trivial descriptions, such as a basic cognitive cycle from inputs to outputs, were not easy to locate for most cognitive architectures. Sometimes, it was possible to piece together a coherent explanation from multiple sources, but for many projects there was simply not enough information. For example, diagrams commonly used to illustrate the structure and processing of the architectures differ greatly in resolution and level of abstraction. While most specified the connections between the modules and

their direction, the contents, volume, or temporal order of the information flow was almost always omitted. The accompanying text did not go deep into the details of the diagrams, either.

We often found descriptions of abilities that were supported in theory but only partially implemented or not implemented at all. In general, unless specifically mentioned by the authors, the status of features was difficult to assess based on publications alone, especially if no quantitative tests were presented as proof.

Usability

We found that available source code of cognitive architectures was accompanied by instructions that allowed users with basic programming knowledge to run a simple demo. However, anything beyond, such as reproducing published results or building a custom agent, usually required much more effort and skill. A part of the problem is that no common frameworks or programming languages emerged for development of cognitive architectures. The implementations we found were equally split between Lisp, Java, Python, and C/C++, and used a wide variety of custom and off-the-shelf libraries.

Further complicating matters is each project's complex and unique structure, often formulated using specialized terminology and implemented in a codebase that is, with few exceptions, not professionally developed and maintained. But even for the projects for which ample support, documentation, and workshops are available, like ACT-R and Soar, customizing the code or reproducing the results are still not easy tasks.

The complexity of the code and lack of commonalities across implementations reinforce the insular nature of most cognitive architectures, acting as barriers for contributions outside the core development groups. This is in stark contrast with the current situation in AI, where the proliferation of platforms, code, documentation, demos, and tutorials significantly democratized access to research tools.

12.1.4 Definitions

Last but not least is the issue of definitions. Fundamental concepts, such as cognition, intelligence, learning, reasoning, etc., have numerous definitions and interpretations at different levels of abstraction. This places cognitive architectures in a precarious spot. On one hand, they must operate with inherently fuzzy and difficult to define concepts. On the other, cognitive architectures are not purely theoretical constructs; implementation leaves no place for ambiguities, even if the concept in question is not well-studied or is ill-defined.

To address this issue, we attempted to distill the meanings of key terms and their interpretations within cognitive architectures at the beginning of each chapter of this book. However, we found that the literature was generally opaque with regard to specific theoretical and terminological commitments. For example, a number of cognitive architectures explore reasoning but omit even the most basic definition of what reasoning is or references to definitions in the literature. One can only make an educated guess based on the specification (itself often incomplete), trace references to similar projects, or refer to the implementation (if source code is available).

Commitment to a certain theory or set of ideas on theoretical and algorithmic levels is one of the reasons for developing cognitive architectures and their biggest advantage. However, when not clearly expressed, these commitments are quite difficult to extract from the literature, especially since they tend to change as the cognitive architecture itself evolves.

12.2 Future directions

In this final section, we discuss open problems that could shape the field of cognitive architectures research moving forward. In our view, the key to progress lies in maintaining and strengthening the collaboration between the fields of cognitive architectures, cognitive science, and AI. Throughout the book, we have consistently highlighted the inherent connection between these disciplines. It is our hope that future research in cognitive architectures will continue to serve as the interdisciplinary bridge, fostering mutual benefits and innovation.

Cognitive architectures by their design combine findings from cognitive science and AI to design blueprints for structure and decomposition of human cognitive abilities and their algorithmic representation, respectively. In return, cognitive architectures offer a way to advance theory and applications by revealing gaps in knowledge, testing hypotheses, making predictions, and identifying algorithm limitations. Lastly, cognitive architectures can provide an alternative to mainstream AI to alleviate the danger of epistemic monocultures. While some of these processes have already been happening organically, even more benefits could be gained from an intentional approach. In addition to interdisciplinary ties, moving forward, it is important to establish and follow better research practices.

12.2.1 General best practices

How does one go about building a cognitive architecture from scratch? Unfortunately, there is no straightforward answer to this question because every cognitive architecture in existence is a one-of-the-kind hand-built artifact. However, there are general principles that can be drawn from other domains and learned from the literature that can guide the process.

Research vs. engineering

To begin, one needs to define a high-level purpose for the project. Generally, cognitive architectures can be divided into those that pursue advancing the theory of human mind and those that aim at solving a practical problem. Depending on this initial choice, the strategies for further exploration and implementation will differ (Vernon, 2016).

Theoretical architectures start with a hypothesis or a theory to be tested. These can arise from various source. Comparing influential theories to find commonalities and differences is one way of establishing a basis of an architecture. Estes (1991) discusses how this can be done using the association and trace theories of memory as an example and what testable questions arise in the process: one can ask whether memory is distributed, what the built-in

memory structures are (if any), and whether a unified architecture for memory exists.

New insights into human abilities and limitations unexplained by current theories can also come from experimental data. Similarly, new experimental methods may provide new levels of detail or new information that was not captured before. For instance, the invention of fMRI thirty years ago provided a new level of fidelity for studying the brain and inspired a number of novel findings and new interpretations of the existing approaches (Bandettini, 2012). Among other things, this technology fundamentally altered our understanding of memory, reward circuitry, and brain plasticity (Rosen and Savoy, 2012).

An engineering approach to modeling the mind begins with a concrete problem to be solved, which can be turned into requirement specifications describing the task, resources, and domain. While in principle, it might be possible to find a solution directly from task specification, theory and human data can also be helpful. For instance, observations of human experts may provide a good starting point into the task and specific areas where humans excel or struggle.

When the high-level theoretical foundation is determined, one needs to implement it to empirically test theory or practical utility. One can start from scratch or build the system from existing mechanisms, modules, or even repurpose existing cognitive architectures. Computational approaches designed from other areas can be imported as well. For example, as mentioned in Section 2.2, production systems were not designed to model cognition initially but turned out to be good representations for cognitive processes.

Although it is useful to separate theoretical and applied aspects of developing cognitive architectures conceptually, in practice, both proceed simultaneously. Overall, building an architecture is a necessarily iterative process that takes a lot of experimentation. Whether one starts with an abstract idea or a concrete task specification, the full extent of the problem is almost never known in advance. When gaps in the knowledge are encountered, it is often infeasible to wait until they are closed. Therefore, building an architecture involves a series of decisions and assumptions that reflect the expertise, preferences, and biases of the developers. Some of these decisions will be motivated by theory, while others will be dictated by the demands of the task or available resources. Some decisions may turn out to be correct, while others may be disproved theoretically or discovered empirically later.

In addition, there are high-level principles that apply to both theoretical and engineering approaches, such as the desiderata discussed in Section 1.2.2. Regardless of the chosen path, developers often aim at parsimony by designing the architecture to satisfy as many conditions as possible with the smallest set of mechanisms. This is virtually impossible to get at the first try. Thus, all cognitive architectures go through multiple partial implementations or versions of the same core structure. This can get messy very quickly, therefore following robust methods for theory formulation, software development, and comprehensive evaluation is very important. The good news is that progress along one of these problems will bring improvements along others. The outline we give below is only a sketch. A forthcoming volume (Ritter, accepted 2022) will further detail every step of building cognitive models, including theory, task selection, implementation, evaluation, and dissemination of the results.

Theory and specification

Historically, the literature on cognitive architectures emphasized conceptual aspects over implementation. But even despite this imbalance, the exact theoretical stance of many cognitive architectures is still difficult to pinpoint. This is further complicated by two factors. First, there are numerous changes to both theory and implementations that most cognitive architectures undergo throughout their lifetime. Second, even if one wants to be explicit about choices, the limits of publication venues in terms of topics and volume may prevent it. Luckily, there are solutions to both problems.

Surveys are primarily useful for providing a snapshot of the state of the field and qualitative comparisons along different dimensions that clarify broad conceptual differences between projects. However, even the most diligent reviewer does not have the same level of understanding as the authors and may be biased in their selection of projects to survey. Thus, a promising option is to involve the architecture developers in the review process, as was done by Samsonovich (2010). A questionnaire was sent to the authors of multiple cognitive architectures to get their take on what aspects of the architectures were implemented and how. Both kinds of review if conducted periodically could help keep track of field development.

The second issue may be in part resolved by the authors as well. As is already done for several prominent architectures, program papers or books that trace development and summarize accomplishments of the individual projects are quite valuable. Such publications detail history, theoretical foundations, cognitive cycle of the architecture, list specific applications, and summarize the development status, outlining which features are fully or partially implemented and which are supported in principle.

In addition to traditional publication routes via journals, conference papers, and books, other publication options should be considered as well. In our experience, deeper technical discussions and personal perspectives expressed in manuals, blogs, white papers, PhD dissertations, and Master's theses were invaluable resources for a better understanding of the motivations and technical challenges behind many cognitive architectures.

Software development

Although all cognitive architectures we reviewed have a substantial software component, there is relatively little discussion of the best software development practices tailored for projects of this complexity and scale.

Recommendations by architecture developers reiterate the importance of a standard development cycle, use of appropriate tools, graphical user interfaces, code refactoring, but especially emphasize hierarchical decomposition of the system into modules and common lower level objects to encourage code reuse (Nii, 1986; Young, 1993; Hayes-Roth, 1996; Barbera et al., 2002; Boicu et al., 2004; Lane and Gobet, 2012). Testing should also be part of development and performed at every level. Lane and Gobet (2012) give a detailed description of such multilevel testing methodology:

- *Unit tests* ensure correctness of the basic implementation (e.g. average is computed correctly from the given inputs);
- *Process tests* validate the program on a functional level (e.g. verify that the chosen activation function in the perceptron is followed). Testing on this

level is decoupled from the implementation, meaning that it is not specific with respect to the programming language or the algorithm. However, such tests are tied to the theory and can even serve as a descriptive specification;
- *Behavioral tests* verify the behavior of the system as a whole.

Another approach to streamlining development is the use of specialized frameworks. For instance, ROS (Quigley et al., 2009) and OpenCV (Bradski, 2000) have become a standard for developing robotic and computer vision applications, respectively. The advantage of these frameworks is that they provide a large set of ready-to-use data structures and components with performance and correctness guarantees, accompanied by extensive documentation. In addition, a large user community that formed around these frameworks helps identify bugs and suggest new features for future releases.

There have been numerous attempts to build similar frameworks for designing cognitive architectures with the help of graphical user interface, low-level data structures, and high-level modules from which more complex systems could be constructed. However, most remained limited to specific kinds of agents or architectures. For example, Jadex (Pokahr et al., 2005) provided tools for creating BDI agents, RoboComp (Manso et al., 2010) was part of the RoboCog architecture and its successor CORTEX, OpenCog (Hart and Goertzel, 2008) mainly supported development of the CogPrime architecture, Nengo targeted models based on the SPA architecture (Bekolay et al., 2014), and Cognitive Systems Toolkit (Paraense et al., 2016) was primarily used for building the MECA cognitive architecture.

Another issue is designing instances of the specific cognitive architecture. Here, efforts have been made to reduce the load on modelers by providing them with high-level descriptive languages. These languages allow specifying high-level behaviors and include compilers that translate human-readable descriptions into executable code. Such projects have been proposed for Soar (Cooper et al., 1996; Crossman et al., 2004a), ACT-R (Salvucci and Lee, 2003), and Clarion (Nasir et al., 2022).

The field of AI faces many similar challenges but on an even larger scale, especially recently. Potential solutions, such as frameworks for designing complex systems and keeping track of model and data versions, have emerged (Martínez-Fernández et al., 2022). Many of these lessons learned would be useful for cognitive architecture researchers as well.

Multilevel evaluation

The field of cognitive architectures has made substantial progress in terms of theoretical findings and implementations. However, the current state of affairs is such that most of the architectures and their instances exist in isolation, making it unclear what they can do or represent. This situation has implications for both theoretical development and computational modeling.

Currently, the only signal from the literature regarding cognitive architecture design is the popularity of certain mechanisms, like blackboard and production systems for memory or neural networks for perception and motor control. Consequently, there is a general tendency to follow more popular theories and implementations when building cognitive architectures, while overlooking alternative options. To correct this bias, it is necessary to establish a set of objective criteria and evaluation protocols for comparisons.

There are obvious technical challenges that make it difficult to establish a robust evaluation framework that would fit most cognitive architectures. The diversity of the implementations and focus on different aspects of cognition can be significant barriers. However, there are also quite a few commonalities across cognitive architectures, at least in terms of the basic components and core abilities that can be tested in an implementation-agnostic way.

Ideally, proper evaluation should target every aspect of cognitive architectures using a combination of qualitative analysis, software testing techniques mentioned earlier, benchmarking, subjective evaluation, user testing, and competitions. All of these have already been sparingly applied to individual architectures (see Section 10.1), but more work is clearly needed. The benefits of broader comparative evaluation are obvious and will be very much worth the effort. Besides objective evaluation, it will open up the possibility of meta-analysis, tracking the improvements of the individual cognitive architectures across versions, and measuring overall progress in the field.

Here too, experience accumulated in the field of computer science may be useful. This includes adoption of methods for software verification and validation, as well as making use of benchmarks, metrics, and evaluation suites created for testing deep learning algorithms on a variety of tasks. While most would not be directly applicable to cognitive architectures, they can serve as a starting point. One example is the Brain-Score benchmark that provides an evaluation framework that uses behavioral and neural data to evaluate vision algorithms at different levels on a set of common vision problems (Schrimpf et al., 2018).

Code and data sharing

At present, public code repositories, models, and documentation are available only for major cognitive architectures, such as ACT-R, Soar, ART, OpenCog, NARS, CHREST, CLARION, and SPA. Given the clear benefits of transparent research practices that include code and data sharing for the field overall and for the researchers themselves, we hope to see more implementations of cognitive architectures become publicly available.

Furthermore, it would be useful to establish reference implementations for classic concepts (e.g. blackboard, semantic networks), data structures (e.g. frames), experiment setups, and system components that are common across architectures. Doing so could promote better development practices and facilitate comparisons between different architectures.

Resources about cognitive architectures

Exposure to past work is another essential ingredient for scientific progress. However, keeping track of past and new research is becoming more difficult due to the rapidly growing volume of scientific literature. Interdisciplinary fields such as cognitive architectures face additional challenges because they unite diverse groups of scientists who do not use the same terminology, approach problems differently, and publish in different venues.

Throughout the history of cognitive architectures, there have been numerous attempts at organizing this field through surveys, reports, and online repositories that familiarize researchers within and outside the cognitive architecture community with the past and ongoing work. We hope to contribute

to this by making the materials for this book available,[2] which we hope will elicit useful feedback from the community as well.

12.2.2 Open research areas

Lastly, we would like to point out challenging problems that could motivate the next generation of researchers and architecture developers. Of course, the list below is incomplete and reflects our own interests and experiences. However, it is not arbitrary and aims to build upon theoretical advances and concrete implementations that have been accumulated across cognitive architectures we examined in this book. Thus, we focus on the broad problems that combine multiple cognitive abilities and as such are uniquely suited for further exploration in cognitive architectures.

Perception and embodiment. One of the important but overlooked aspects of cognition is the connection between body and perception. It is well known that perceptual and motor abilities of living organisms are linked to the structure, dimensions, and kinematics of their bodies (Vernon et al., 2016). For example, in humans, optical properties of the eyes and visual system operation are inseparable, thus any artificial system that aims at explaining human vision should respect this connection. For future vision research this means using hardware that closely mimics optical, physical, and kinetic properties of human head and eyes. Similarly, visual processing algorithms should take into account available sensor parameters explicitly.

Besides vision, considering other senses and interaction between them will have both theoretical and practical significance. The theoretical questions concern the mechanisms and features used for integrating, translating, and coordinating percepts across different senses. In practice, effective combination of visual, auditory, and touch modalities will be highly desirable for social and industrial applications.

Learning and memory. Humans learn in many ways, from observation, by trial and error, through instruction, by induction, and so on. Some learned information is then retained in memory. While individual forms of learning and memory have been extensively studied in psychology, and numerous computational models exist in cognitive architectures, many puzzles remain unsolved.

Working memory is recognized as a crucial component of human cognition, but there is still no consensus on its properties, such as location, structure, contents, limits, function, and interaction with other types of memory. Regarding learning, potential directions for investigation would focus on the mechanisms that guide selection of a specific learning mode depending on the context, how learning occurs for different types of knowledge, and what controls when learning starts and ends.

Lifelong learning is yet another mystery, but long-term accumulation and consolidation of knowledge is difficult to study within existing cognitive architectures. Expanding the lifespan of these systems will be necessary to explain how human experts obtain their knowledge or skills after thousands of hours of learning spread over years.

Multitasking. It would be an understatement to say that the human brain is versatile. Even in theory, the number of tasks and abilities that humans

[2]

possess has not been exhaustively enumerated. While cognitive architectures as a whole cover a non-negligible portion of the human cognitive phenomena discovered so far, there are very few systems that can demonstrate even a handful of distinct abilities within a single instance. Moving from single-phenomenon or single-task models to more general ones is a big leap, likely more challenging than putting together existing single-task models. The interaction between different task requirements and potential conflicts between mechanisms that implement them will undoubtedly complicate both development and evaluation. Nevertheless, some interesting areas for further investigation are the mechanisms that select and tune processing depending on the task demands and context, resource allocation, interruption handling, and concurrent execution of several tasks.

Attention. Attention, much like perception, has been largely neglected in cognitive architectures but remains a very active area of research in cognitive science. Theories and empirical data generated in the last hundred or so years of research implicate attention in virtually all task-driven behavior as well as in how perception, memory, and learning operate. As such, attention cannot be feasibly modeled in isolation from other elements of cognition and vice versa. Some specific questions of interest are the nature of connection between attention and working memory, a broader role of attention as a mechanism that inhibits and restricts, rather than a primarily selective process, as well as what attention operates on, how, and in what contexts.

Developmental aspects of cognition. There are still very few attempts to understand how the human brain develops and ages. Cognitive architectures and AI so far predominantly focused on modeling the adult brain and as such maintain a fixed structure, whereas the human brain shows incredible plasticity both structurally and functionally throughout its lifetime. Discovering the set of innate structures and mechanisms that give rise to the general and robust visual, learning, and reasoning abilities in adults is one of the pressing problems.

12.3 Conclusion

Some readers may think that cognitive architectures have become irrelevant in the age of deep learning. We argue that while modern deep learning AI is a viable solution for practical applications, it is likely not the solution that cognitive architectures seek. As we discussed earlier in Chapter 11, mainstream AI is drifting away from its already loose ties to human biology; the resulting black boxes trained on inconceivably large amounts of data using enough energy to melt a glacier are not comparable to how the human brain learns and operates. Even if human-like performance is achieved with the current or future version of deep learning methods, the mysteries of the human mind will remain unsolved.

Peering into the depths of the human cognition to discover its mechanisms is not an easy task. Building a cognitive architecture is a major commitment that requires significant human and computational resources, interdisciplinary expertise, and time to put together even the most basic working prototype. Not surprisingly, the average lifespan of the cognitive architectures in our sample is more than a decade. This is what it takes to develop theoretical foundations

and software implementation to the point where the cognitive architecture can model non-trivial phenomena, produce insights, and perform interesting tasks.

Tremendous amounts of work have been done already, and even more opportunities lie ahead. Each open problem outlined in this chapter can inspire dozens of research projects. Regardless of the chosen research direction, we believe that success of future explorations in cognitive architectures will be achieved with the help of robust development practices. Specifically, improvements in the following four areas are most needed to boost further development of the field: 1) focus on more realistic perceptual systems with tighter integration between attention, reasoning, and action, 2) robust quantitative evaluation procedures based on publicly available data (including the results of human experiments), 3) availability of implementations and precise technical specifications, and 4) clarity of definitions and theoretical commitments.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems, *arXiv:1603.04467*.

Abbott, L. F. and Kepler, T. B. (2005). Model neurons: From Hodgkin-Huxley to Hopfield, *in* L. Garrido (ed.), *Statistical Mechanics of Neural Networks*, Springer, pp. 5–18.

Aben, B., Stapert, S., and Blokland, A. (2012). About the distinction between working memory and short-term memory, *Frontiers in Psychology* **3**: 301.

Abraham, T. H. (2002). (Physio) logical circuits: The intellectual origins of the McCulloch–Pitts neural networks, *Journal of the History of the Behavioral Sciences* **38**(1): 3–25.

Adams, S. S., Banavar, G., and Campbell, M. (2016). I-athlon: Towards a multidimensional Turing test, *AI Magazine* **37**(1): 78–84.

Agre, P. E. and Chapman, D. (1990). What are plans for?, *Robotics and Autonomous Systems* **6**(1–2): 17–34.

Ahmed, N., Wahed, M., and Thompson, N. C. (2023). The growing influence of industry in AI research, *Science* **379**(6635): 884–886.

Aiyappa, R., An, J., Kwak, H., and Ahn, Y.-Y. (2023). Can we trust the evaluation on ChatGPT?, *Proceedings of the Workshop on Trustworthy Natural Language Processing*, pp. 47–54.

Alasehir, O. and Acarturk, C. (2022). Interdisciplinarity in cognitive science: A document similarity analysis, *Cognitive Science* **46**(12): e13222.

Albus, J. and Barbera, A. (2006). Intelligent control and tactical behavior development: A long term NIST partnership with the Army, *Proceedings of the International Joint Topical Meeting on Emergency Preparedness & Response and Robotics & Remote Systems (EPRRSD)*, pp. 1–11.

Albus, J. et al. (2002). 4D/RCS: A reference model architecture for unmanned vehicle systems version 2.0, *Technical Report 6910*, National Institute of Standards and Technology.

Albus, J., Bostelman, R., Chang, T., Hong, T., Shackleford, W., and Shneier, M. (2006). Learning in a hierarchical control system: 4D/RCS in the DARPA LAGR program, *Journal of Field Robotics* **23**(11–12): 975–1003.

Albus, J. S. (1991). A theory of intelligent machine systems, *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, pp. 3–9.

Albus, J. S. (1994). A reference model architecture for intelligent systems design, *Technical Report 5502*, NIST.

Albus, J. S. (1996). The NIST Real-Time Control System (RCS): A reference model architecture for computational Intelligence, *Proceedings of the Workshop on Computational Intelligence and Their Impact on Future High Performance Engineering Systems*, pp. 23–42.

Albus, J. S. (1997). The NIST real-time control system (RCS): An approach to intelligent systems research, *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2–3): 157–174.

Albus, J. S. (2002). 4D/RCS: A reference model architecture for intelligent unmanned ground vehicles, *Unmanned Ground Vehicle Technology IV*, Vol. 4715, International Society for Optics and Photonics, pp. 303–310.

Albus, J. S. and Barbera, A. J. (2005). RCS: A cognitive architecture for intelligent multi-agent systems, *Annual Reviews in Control* **29**(1): 87–99.

Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A. I., Babaei, H., LeJeune, D., Siahkoohi, A., and Baraniuk, R. G. (2024). Self-consuming generative models go MAD, *Proceedings of the International Conference on Learning Representations*.

Aler, R., Borrajo, D., and Fernández, S. (2003). On providing prior knowledge for learning relational search heuristics, *Proceedings of the Workshop on Planning, Scheduling and Temporal Reasoning*.

Alexander, P. A., Schallert, D. L., and Reynolds, R. E. (2009). What is learning anyway? A topographical perspective considered, *Educational Psychologist* **44**(3): 176–192.

Alfonseca, M. (1989). Frames, semantic networks, and object-oriented programming in APL2, *IBM Journal of Research and Development* **33**(5): 502–510.

Allen, C. (2017). On (not) defining cognition, *Synthese* **194**(11): 4233–4249.

Allen, J. and Sun, R. (2016). Emotion contagion in a cognitive architecture, *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp.

Allender, L. (2000). Modeling human performance: Impacting system design, performance, and cost, *Simulation Series* **32**(3): 139–144.

Almaatouq, A., Griffiths, T. L., Suchow, J. W., Whiting, M. E., Evans, J., and Watts, D. J. (2024). Beyond playing 20 questions with nature: Integrative experiment design in the social and behavioral sciences, *Behavioral and Brain Sciences* **47**: e33.

ALPAC (1966). Language and machines: Computers in translation and linguistics, *Technical Report 1416*, National Academy of Sciences.

Altmann, E. M. and Gray, W. D. (2000). An integrated model of set shifting and maintenance, *Proceedings of the International Conference on Cognitive Modeling*, pp. 17–24.

Alvarez, M. (2010). Reasons for action and practical reasoning, *Ratio* **23**(4): 355–373.

Amsel, A. (1992). B. F. Skinner and the Cognitive Revolution, *Journal of Behavior Therapy and Experimental Psychiatry* **23**(2): 67–70.

Anderson, J. R. (1983a). *The Architecture of Cognition*, Psychology Press.

Anderson, J. R. (1983b). Retrieval of information from long-term memory, *Science* **220**(4592): 25–30.

Anderson, J. R. (1991). Is human cognition adaptive?, *Behavioral and Brain Sciences* **14**(3): 471–485.

Anderson, J. R. and Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval, *Journal of Experimental Psychology: Learning, Memory, and Cognition* **27**(6): 1331.

Anderson, J. R. and Lebiere, C. (2003). The Newell test for a theory of cognition, *Behavioral and Brain Sciences* **26**(5): 587–601.

Anderson, J. R. and Lebiere, C. J. (1998). *The Atomic Components of Thought*, Psychology Press.

Anderson, J. R. and Schooler, L. J. (1991). Reflections of the environment in memory, *Psychological Science* **2**(6): 396–408.

Anderson, J. R., Reder, L. M., and Lebiere, C. (1996). Working memory: Activation limitations on retrieval, *Cognitive Psychology* **30**(3): 221–256.

Anderson, J. R., Bothell, D., Lebiere, C., and Matessa, M. (1998). An integrated theory of list memory, *Journal of Memory and Language* **38**(4): 341–380.

Anderson, J. R., Qin, Y., Sohn, M.-H., Stenger, V. A., and Carter, C. S. (2003). An information-processing model of the BOLD response in symbol manipulation tasks, *Psychonomic Bulletin & Review* **10**(2): 241–261.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind, *Psychological Review* **111**(4): 1036–1060.

Anderson, J. R., Zhang, Q., Borst, J. P., and Walsh, M. M. (2016). The discovery of processing stages: Extension of Sternberg's method, *Psychological Review* **123**(5): 481.

Anderson, M. L. and Oates, T. (2007). A review of recent research in metareasoning and metalearning, *AI Magazine* **28**(1): 12–12.

Andrews, R., Diederich, J., and Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems* **8**(6): 373–389.

Aradhye, H. B., Bakshi, B. R., Davis, J. F., and Ahalt, S. C. (2004). Clustering in wavelet domain: A multiresolution ART network for anomaly detection, *Journal of the Amrtical Institute of Chemical Engineers* **50**(10): 2455–2466.

Arbib, M. (1961). Turing machines, finite automata and neural nets, *Journal of the ACM* **8**(4): 467–475.

Arel, I., Rose, D., and Karnowski, T. (2009). A deep learning architecture comprising homogeneous cortical circuits for scalable spatiotemporal pattern inference, *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS) Workshop on Deep Learning for Speech Recognition and Related Applications*, pp. 23–32.

Arifovic, J. (2005). The implementation of the Turing Tournament: A report, *in* T. Lux, S. Reitz, and E. Samanidou (eds), *Nonlinear Dynamics and Heterogeneous Interacting Agents*, Springer, pp. 3–9.

Arkin, R. C. (1989). Motor schema-based mobile robot navigation, *International Journal of Robotics Research* **8**(4): 92–112.

Aron, J. (2011). Software tricks people into thinking it is human. `https://www.newscientist.com/article/dn20865-software-tricks-people-into-thinking-it-is-human`. Accessed May 2, 2025.

Arrabales, R., Ledezma, A., and Sanchis, A. (2009a). Establishing a roadmap and metrics for conscious machines development, *Proceedings of the IEEE International Conference on Cognitive Informatics*, pp. 94–101.

Arrabales, R., Ledezma, A., and Sanchis, A. (2009b). Towards conscious-like behavior in computer game characters, *Proceedings of the Symposium on Computational Intelligence and Games*, pp. 217–224.

Arrabales, R., Ledezma Espino, A. I., and Sanchis de Miguel, M. A. (2009c). CERA-CRANIUM:

A test bed for machine consciousness research, *Proceedings of the International Workshop on Machine Consciousness*.

Arrabales, R., Ledezma Espino, A. I., and Sanchis de Miguel, M. A. (2009d). A cognitive approach to multimodal attention, *Journal of Physical Agents* **3**(1): 53–64.

Arrabales, R., Ledezma, A., and Sanchis, A. (2011). Simulating visual qualia in the CERA-CRANIUM cognitive architecture, *in* C. Hernández, J. Goméz-Ramirez, R. Sanz, L. S. Smith, A. Hussain, A. Chella, and I. Aleksander (eds), *From Brains to Systems*, Springer, pp. 223–238.

Artal, P. (2015). Image formation in the living human eye, *Annual Review of Vision Science* **1**: 1–17.

Aryananda, L. (2001). Online and unsupervised face recognition for humanoid robot: Toward relationship with people, *Proceedings of the International Conference on Humanoid Robots*.

Asensio, J. M. L., Peralta, J., Arrabales, R., Bedia, M. G., Cortez, P., and Peña, A. L. (2014). Artificial intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters, *Expert Systems with Applications* **41**(16): 7281–7290.

Atkins, E. M. and Durfee, E. H. (1997). Development of iterative real-time scheduler to planner feedback, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1267–1272.

Atkins, E. M., Durfee, E. H., and Shin, K. G. (1997). Buying time for resource-bounded planning, *Proceedings of the AAAI Workshop on Building Resource-Bounded Reasoning Systems*, pp. 7–11.

Atkinson, R. C. and Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes, *Psychology of Learning and Motivation*, Vol. 2, Elsevier, pp. 89–195.

Austin, T. and Stokey, R. (2011). Integrated mission, vehicle, and sensor control of the iPUMA AUV, *Technical report*, Woods Hole Oceanographic Institution.

Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain, *Journal of Comparative Neurology* **513**(5): 532–541.

Baars, B. J. (1983). Conscious contents provide the nervous system with coherent, global information, *in* R. J. Davidson, G. E. Schwartz, and D. Shapiro (eds), *Consciousness and Self-Regulation*, Vol. 3, Springer Science & Business Media, LLC, pp. 41–79.

Baars, B. J. (1988). *A Cognitive Theory of Consciousness*, Cambridge University Press.

Baars, B. J. and Franklin, S. (2003). How conscious experience and working memory interact, *Trends in Cognitive Sciences* **7**(4): 166–172.

Bach, J. (2009). *Principles of Synthetic Intelligence PSI: An Architecture of Motivated Cognition*, Oxford University Press.

Bach, J. (2011). Modeling emotion as an interaction between motivation and modulated cognition, *Proceedings of the Workshop on Standards in Emotion Modeling*.

Bach, J. (2012a). A framework for emergent emotions, based on motivation and cognitive modulators, *International Journal of Synthetic Emotions* **3**(1): 43–63.

Bach, J. (2012b). Modeling motivation and the emergence of affect in a cognitive agent, *in* P. Wang and B. Goertzel (eds), *Theoretical Foundations of Artificial General Intelligence*, Atlantis Press, pp. 241–262.

Bach, J. (2015). Modeling motivation in MicroPsi 2, *International Conference on Artificial General Intelligence*, Springer, pp. 3–13.

Bachiller, P., Núnez, P., and Bandera, A. (2021). DSR_d: A Proposal for a low-latency, distributed working memory for CORTEX, *Proceedings of the International Workshop of Physical Agents*, pp. 109–122.

Baddeley, A. (1986). *Working Memory*, Oxford University Press.

Baddeley, A. (2000). The episodic buffer: A new component of working memory?, *Trends in Cognitive Sciences* **4**(11): 417–423.

Baddeley, A. (2012). Working memory: Theories, models, and controversies, *Annual Review of Psychology* **63**: 1–29.

Baddeley, A. D. and Hitch, G. (1974). Working memory, *in* G. H. Bower (ed.), *Psychology of Learning and Motivation*, Vol. 8, Elsevier, pp. 47–89.

Bader, S. and Hitzler, P. (2005). Dimensions of neural-symbolic integration—A structured survey, *in* S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, and J. Woods (eds), *We Will Show Them: Essays in Honour of Dov Gabbay*, King's College Publications, pp. 167–194.

Bagchi, S. and Kawamura, K. (1992). An architecture of a distributed object-oriented robotic system, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 711–716.

Bakan, D. (1966). The test of significance in psychological research, *Psychological Bulletin* **66**(6): 423–437.

Baker, R. S., Corbett, A. T., and Koedinger, K. R. (2003). Statistical techniques for comparing ACT-R models of cognitive performance, *Proceedings of the Annual ACT-R Workshop*, pp. 129–134.

Bandera, A., Bandera, J. P., Bustos, P., Férnandez, F., García-Olaya, A., García-Polo, J., García-Varea, I., Manso, L. J., Marfil, R., Martínez-Gómez, J., et al. (2017). LifeBots I: Building the software infrastructure for supporting lifelong technologies, *Proceedings of the Iberian Robotics Conference*, pp. 391–402.

Bandettini, P. A. (2012). Twenty years of functional MRI: The science and the stories, *NeuroImage* **62**(2): 575–588.

Bao, P., She, L., McGill, M., and Tsao, D. Y. (2020). A map of object space in primate inferotemporal cortex, *Nature* **583**(7814): 103–108.

Barber, D. (2012). *Bayesian Reasoning and Machine Learning*, Cambridge University Press.

Barbera, T., Messina, E., Huang, H.-M., Schlenoff, C., and Balakirsky, S. (2002). Software engineering for intelligent control systems, *Technical Report 7095*, National Institute of Standards and Technology.

Barbu, A., Mayo, D., Alverio, J., Luo, W., Wang, C., Gutfreund, D., Tenenbaum, J., and Katz, B. (2019). ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models, *Advances in Neural Information Processing Systems* **32**: 9453–9463.

Barrow, H., Tenenbaum, J., Hanson, A., and Riseman, E. (1978). Recovering intrinsic scene characteristics, *in* A. Hanson and E. Riseman (eds), *Computer Vision Systems*, pp. 3–26.

Barsalou, L. W. (1999). Perceptual symbol systems, *Behavioral and Brain Sciences* **22**(4): 577–660.

Bartol Jr, T. M., Bromer, C., Kinney, J., Chirillo, M. A., Bourne, J. N., Harris, K. M., and Sejnowski, T. J. (2015). Nanoconnectomic upper bound on the variability of synaptic plasticity, *Elife* **4**: e10778.

Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G. E., and Lillicrap, T. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures, *Advances in Neural Information Processing Systems*, pp. 9368–9378.

Barwich, A.-S. (2019). The value of failure in science: The story of grandmother cells in neuroscience, *Frontiers in Neuroscience* **13**: 1121.

Bateson, P. and Mameli, M. (2007). The innate and the acquired: Useful clusters or a residual distinction from folk biology?, *Developmental Psychobiology* **49**(8): 818–831.

Bayne, T., Brainard, D., Byrne, R. W., Chittka, L., Clayton, N., Heyes, C., Mather, J., Ölveczky, B., Shadlen, M., Suddendorf, T., and Webb, B. (2019). What is cognition?, *Current Biology* **29**(13): R608–R615.

Beal, J., Wu, H.-Y., Park, D. H., Zhai, A., and Kislyuk, D. (2022). Billion-scale pretraining with vision transformers for multi-task visual representations, *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 564–573.

Beddiar, D. R., Nini, B., Sabokrou, M., and Hadid, A. (2020). Vision-based human activity recognition: a survey, *Multimedia Tools and Applications* **79**: 30509–30555.

Bekey, G. A. (1996). Biologically inspired control of autonomous robots, *Robotics and Autonomous Systems* **18**(1-2): 21–31.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., Choo, X., Voelker, A. R., and Eliasmith, C. (2014). Nengo: A Python tool for building large-scale functional brain models, *Frontiers in Neuroinformatics* **7**: 48.

Bell, M. Z. (1985). Why expert systems fail, *Journal of the Operational Research Society* **36**(7): 613–619.

Bellas, F. and Duro, R. J. (2003). Introducing long term memory in an ANN based Multilevel Darwinist Brain, *in* J. Mira and J. R. Álvarez (eds), *Computational Methods in Neural Modeling*, Springer, pp. 590–597.

Bellas, F., Becerra, J. A., and Duro, R. J. (2006). Some experimental results with a two level memory management system in the Multilevel Darwinist Brain, *Proceedings of the European Symposium on Artificial Neural Networks*, pp. 113–118.

Bellas, F., Duro, R. J., Faiña, A., and Souto, D. (2010a). Multilevel Darwinist Brain (MDB): Artificial evolution in a cognitive architecture for real robots, *IEEE Transactions on Autonomous Mental Development* **2**(4): 340–354.

Bellas, F., Faiña, A., Varela, G., and Duro, R. J. (2010b). A cognitive developmental robotics architecture for lifelong learning by evolution in real robots, *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1–8.

Bello, P., Bridewell, W., and Wasylyshyn, C. (2016). Attentive and pre-attentive processes in multiple object tracking: A computational investigation, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Bender, A. (2019). The value of diversity in cognitive science, *Topics in Cognitive Science* **11**(4): 853–863.

Bender, E. M., Gebru, T., McMillan-Major, A., and Mitchell, M. (2021). On the dangers of stochastic parrots: Can language models be too big?, *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623.

Bengio, Y. (2009). Learning deep architectures for AI, *Foundations and Trends® in Machine Learning* **2**(1): 1–127.

Bengio, Y. (2019a). Definition of deep learning, `https://github.com/MontrealAI/MontrealAI.github.io/blob/master/aidebate/yoshuadefinitiondeeplearning.png`. [Accessed July 5, 2024].

Bengio, Y. (2019b). Response to Gary Marcus, `https://docs.google.com/document/d/1P9YyZ4xEDO98qPTa1Al4RnTqxj3mMiCvQE4i3hZkthY/edit?fbclid=IwAR1BH94ACSru2CGObXPTEdNV7ElGOoUB2WUg5Ko3htK5uwzZiN4YinoE4Xs`. [Accessed July 5, 2024].

Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2017). STDP-compatible approximation of backpropagation in an energy-based model, *Neural Computation* **29**(3): 555–577.

Benjamin, D. P., Lyons, D., and Lonsdale, D. (2004). ADAPT: A cognitive architecture for robotics, *International Conference on Cognitive Modeling*, pp. 337–338.

Benjamin, D. P., Lyons, D., and Achtemichuk, T. (2006). Obstacle avoidance using predictive vision based on a dynamic 3D world model, *Intelligent Robots and Computer Vision XXIV: Algorithms, Techniques, and Active Vision*, Vol. 6384, International Society for Optics and Photonics.

Benjamin, D. P., Funk, C., and Lyons, D. (2013). A cognitive approach to vision for a mobile robot, *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Vol. 8756, International Society for Optics and Photonics, p. 87560I.

Benjamin, D. P., Lyons, D., and Lynch, R. (2015). Effects of using a 3D model on the performance of vision algorithms, *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Vol. 9498, International Society for Optics and Photonics, p. 94980B.

Bennett, C. A. (1971). Toward empirical, practicable, comprehensive task taxonomy, *Human Factors* **13**(3): 229–235.

Berko, J. (1958). The child's learning of English morphology, *Word* **14**(2-3): 150–177.

Berrar, D., Konagaya, A., and Schuster, A. (2013). Turing test considered mostly harmless, *New Generation Computing* **31**(4): 241–263.

Besold, T., Hernández-Orallo, J., and Schmid, U. (2015). Can machine intelligence be measured in the same way as human intelligence?, *Künstliche Intelligenz* **29**(3): 291–297.

Biggs, J. B. (1988). Assessing student approaches to learning, *Australian Psychologist* **23**(2): 197–206.

Block, N. (1981). Psychologism and behaviorism, *Philosophical Review* **90**(1): 5–43.

Bloom, B. S. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*, Longmans.

Blouw, P., Solodkin, E., Thagard, P., and Eliasmith, C. (2016). Concepts as semantic pointers: A framework and computational model, *Cognitive Science* **40**(5): 1128–1162.

Blythe, J., Etzioni, O., Gil, Y., Joseph, R., Kahn, D., Knoblock, C., Minton, S., Pérez, A., Reilly, S., Veloso, M., Wang, X., and Carbonell, J. G. (1992). PRODIGY4.0: The manual and tutorial, *Technical Report CMU-CS-92-150*, Carnegie Mellon University.

Blythe, J., Veloso, M., and de Souza, L. E. (1997). The Prodigy user interface, *Technical Report CMU-CS-97-114*, Carnegie Mellon University.

Bobrow, D. G. (1964). A question-answering system for high school algebra word problems, *Proceedings of the AFIPS Fall Joint Computer Conference*, pp. 591–614.

Boden, M. A. (2006). *Mind as Machine: A History of Cognitive Science*, Oxford University Press.

Boden, M. A. (2010). The Turing test and artistic creativity, *Kybernetes* **39**(3): 409–413.

Boicu, C. and Tecuci, G. (2004). Mixed-initiative ontology learning, *Proceedings of the International Conference on Artificial Intelligence (IC-AI)*, pp. 745–751.

Boicu, M., Tecuci, G., Marcu, D., Bowman, M., Shyr, P., Ciucu, F., and Levcovici, C. (2000). Disciple-COA: From agent programming to agent teaching, *Proceedings of the International Conference on Machine Learning*, pp. 73–80.

Boicu, M., Tecuci, G., Stanescu, B., Marcu, D., Barbulescu, M., and Boicu, C. (2004). Design principles for learning agents, *Proceedings of AAAI Workshop on Intelligent Agent Architectures: Combining the Strengths of Software Engineering and Cognitive Systems*, pp. 26–33.

Bommasani, R. et al. (2021). On the opportunities and risks of foundation models, *arXiv:2108.07258*.

Bonasso, P. R., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., and Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents, *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2-3): 237–256.

Bonasso, R. P. (2001). Intelligent control of a NASA advanced water recovery system, *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Bonasso, R. P. and Kortenkamp, D. (1995). Characterizing an architecture for intelligent, reactive agents, *Proceedings of the AAAI Spring Symposium*, pp. 29–34.

Bonasso, R. P., Kortenkamp, D., and Thronesbery, C. (2003). Intelligent control of a water-recovery system: Three years in the trenches, *AI Magazine* **24**(1): 19–44.

Borrajo, D. and Veloso, M. (1994). Incremental learning of control knowledge for nonlinear problem solving, *Proceedings of the European Conference on Machine Learning*, pp. 64–82.

Borst, J. P., Nijboer, M., Taatgen, N. A., van Rijn, H., and Anderson, J. R. (2015). Using data-driven model-brain mappings to constrain formal models of cognition, *PLoS One* **10**(3): e0119673.

Bothell, D. (2017). ACT-R 7 reference manual, `https://act-r.psy.cmu.edu/actr7/reference-manual.pdf`.

Botvinick, M. M. (2008). Hierarchical models of behavior and prefrontal function, *Trends in Cognitive Sciences* **12**(5): 201–208.

Bowers, J. S. (2017). Grandmother cells and localist representations: A review of current thinking, *Language, Cognition and Neuroscience* **32**(3): 257–273.

Bowers, J. S., Malhotra, G., Dujmović, M., Montero, M. L., Tsvetkov, C., Biscione, V., Puebla, G., Adolfi, F., Hummel, J. E., Heaton, R. F., et al. (2023). Deep problems with neural network models of human vision, *Behavioral and Brain Sciences* **46**: e385.

Bradski, G. (2000). The OpenCV Library, *Dr. Dobb's Journal of Software Tools*.

Bradski, G., Carpenter, G. A., and Grossberg, S. (1992). Working memory networks for learning temporal order with application to three-dimensional visual object recognition, *Neural Computation* **4**(2): 270–286.

Braine, M. D. and O'Brien, D. P. (1991). A theory of if: A lexical entry, reasoning program, and pragmatic principles, *Psychological Review* **98**(2): 182–203.

Bratman, M. (1987). *Intention, Plans, and Practical Reason*, Harvard University Press.

Breazeal, C. (1998a). A motivational system for regulating human-robot interaction, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 54–61.

Breazeal, C. (1998b). Regulating human-robot interaction using 'emotions', 'drives', and facial expressions, *Proceedings of the International Conference on Autonomous Agents*, Vol. 98, pp. 14–21.

Breazeal, C. (2003a). Emotion and sociable humanoid robots, *International Journal of Human-Computer Studies* **59**(1-2): 119–155.

Breazeal, C. (2003b). Toward sociable robots, *Robotics and Autonomous Systems* **42**(3-4): 167–175.

Breazeal, C. and Aryananda, L. (2002). Recognition of affective communicative intent in robot-directed speech, *Autonomous Robots* **12**(1): 83–104.

Breazeal, C. and Scassellati, B. (1999). A context-dependent attention system for a social robot, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1146–1152.

Breazeal, C. and Scassellati, B. (2002). Challenges in building robots that imitate people, *in* K. Dautenhahn and C. L. Nehaniv (eds), *Imitation in Animals and Artifacts*, MIT Press, pp. 363–390.

Breazeal, C., Edsinger, A., Fitzpatrick, P., Scassellati, B., and Varchavskaia, P. (2000). Social constraints on animate vision, *Intelligent Systems and Their Applications* **15**(4): 32–37.

Breazeal, C., Edsinger, A., Fitzpatrick, P., and Scassellati, B. (2001). Active vision for sociable robots, *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* **31**(5): 443–453.

Breiman, L. (2001). Random forests, *Machine Learning* **45**: 5–32.

Bresina, J. and Drummond, M. (1990). Integrating planning and reaction: A preliminary report, *Proceedings of the AAAI Spring Symposium on Planning in Uncertain, Unpredictable or Changing Environments*.

Bridewell, W. and Bello, P. (2015). Incremental object perception in an attention-driven cognitive architecture, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 279–284.

Bridewell, W. and Bello, P. (2016). A theory of attention for cognitive systems, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, pp. 1–16.

Brigandt, I. (2005). The instinct concept of the early Konrad Lorenz, *Journal of the History of Biology* **38**(3): 571–608.

Bringsjord, S. and Schimanski, B. (2003). What is artificial intelligence? Psychometric AI as an answer, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 887–893.

Brom, C. and Bryson, J. (2006). Action selection for intelligent systems. White paper for the European Network for the Advancement of Artificial Cognitive Systems.

Brooks, R. (1986). A robust layered control system for a mobile robot, *Journal on Robotics and Automation* **2**(1): 14–23.

Brooks, R. A. (1985). A mobile robot project, *Technical Report 265*, MIT.

Brooks, R. A. (1987). Planning is just a way of avoiding figuring out what to do next, *Technical Report 303*, MIT.

Brooks, R. A. (1989). Engineering approach to building complete, intelligent beings, *Intelligent Robots and Computer Vision VII*, Vol. 1002, pp. 618–625.

Brooks, R. A. (1990). Elephants don't play chess, *Robotics and Autonomous Systems* **6**(1–2): 3–15.

Brooks, R. A. (1991). Intelligence without representation, *Artificial Intelligence* **47**(1-3): 139–159.

Brooks, R. A. (2014). The role of learning in autonomous robots, *Proceedings of the Annual Workshop on Computational Learning Theory*, pp. 5–10.

Brooks, R. A. and Flynn, A. M. (1989). Fast, cheap and out of control: A robot invasion of the solar system, *Journal of the British Interplanetary Society* **42**: 478–485.

Brooks, R. A., Breazeal, C., Marjanović, M., Scassellati, B., and Williamson, M. M. (1999). The Cog project: Building a humanoid robot, *in* C. Nehaniv (ed.), *Computation for Metaphors, Analogy, and Agents*, Springer-Verlag Heidelberg Berlin, pp. 52–87.

Brown, J. W. (2014). The tale of the neuroscientists and the computer: Why mechanistic theory matters, *Frontiers in Neuroscience* **8**: 349.

Brown, N. and Sandholm, T. (2019). Superhuman AI for multiplayer poker, *Science* **365**(6456): 885–890.

Bruce, N. and Tsotsos, J. (2005). Saliency based on information maximization, *Advances in Neural Information Processing Systems* **18**: 155–162.

Brunel, N., Mark, C., and van Rossum, W. (2007). Quantitative investigations of electrical nerve excitation treated as polarization, *Biological Cybernetics* **97**(5-6): 341.

Bruner, J. S. (1969). Modalities of memory, *in* G. A. Talland and N. C. Waugh (eds), *The Pathology of Memory*, pp. 253–259.

Brysbaert, M. (2019). How many words do we read per minute? A review and meta-analysis of reading rate, *Journal of Memory and Language* **109**: 104047.

Brysbaert, M., Stevens, M., Mandera, P., and Keuleers, E. (2016). How many words do we know? Practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age, *Frontiers in Psychology* **7**: 1116.

Bryson, A. E. and Denham, W. F. (1962). A steepest-ascent method for solving optimum programming problems, *Journal of Applied Mechanics* **29**: 247–257.

Bryson, J. (1999). Hierarchy and sequence vs. full parallelism in action selection, *Intelligent Virtual Agents*, Vol. 2, pp. 113–125.

Buchanan, B., Sutherland, G., and Feigenbaum, E. A. (1969). Heuristic DENDRAL: A program for generating explanatory hypotheses, *in* D. Michie and B. Meltzer (eds), *Machine Intelligence*, pp. 209–254.

Buchanan, B. G. (1986). Expert systems: Working systems and the research literature, *Expert Systems* **3**(1): 32–50.

Burghardt, G. M. (2019). A place for emotions in behavior systems research, *Behavioural Processes* **166**: 103881.

Burns, G. et al. (2014). Reports on the 2013 AAAI Fall Symposium Series, *AI Magazine* **35**(2): 69–74.

Burstein, M. H. (1989). Analogy vs. CBR: The purpose of mapping, *Proceedings of the Workshop on Case-Based Reasoning*, pp. 133–136.

Bustos, P., Martinez-Gomez, J., Garcia-Varea, I., Rodriguez-Ruiz, L., Bachiller, P., Calderita, L., Manso, L., Sanchez, A., Bandera, A., and Bandera, J. (2013). Multimodal interaction with Loki, *Proceedings of the Workshop of Physical Agents*, pp. 53–60.

Bustos, P., Manso, L. J., Bandera, A. J., Bandera, J. P., Garcia-Varea, I., and Martinez-Gomez, J. (2019). The CORTEX cognitive robotics architecture: Use cases, *Cognitive Systems Research* **55**: 107–123.

Byrne, M. D. (2001). ACT-R/PM and menu selection: Applying a cognitive architecture to HCI, *International Journal of Human-Computer Studies* **55**(1): 41–84.

Byrne, M. D., Anderson, J. R., Douglass, S., and Matessa, M. (1999). Eye tracking the visual search of click-down menus, *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 402–409.

Byrnes, R., MacPherson, D., Kwak, S., McGhee, R., and Nelson, M. (1992). An experimental comparison of hierarchical and subsumption software architectures for control of an autonomous underwater vehicle, *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pp. 135–141.

Caglayan, A., Snorrason, M., Jacoby, J., Mazzu, J., Jones, R., and Kumar, K. (1997). Learn Sesame—A learning agent engine, *Applied Artificial Intelligence* **11**(5): 393–412.

Cambria, E., Livingstone, A., and Hussain, A. (2012). The hourglass of emotions, *in* A. Esposito, A. M. Esposito, A. Vinciarelli, R. Hoffmann, and V. C. Müller (eds), *Cognitive Behavioural Systems*, Springer, pp. 144–157.

Campbell, M., Hoane Jr, A. J., and Hsu, F.-h. (2002). Deep Blue, *Artificial Intelligence* **134**(1–2): 57–83.

Camus, T., Coombs, D., Herman, M., and Hong, T.-H. (1996). Real-time single-workstation obstacle avoidance using only wide-field flow divergence, *Proceedings of the International Conference on Pattern Recognition*, Vol. 3, pp. 323–330.

Canaan, R., Salge, C., Togelius, J., and Nealen, A. (2019). Leveling the playing field: Fairness in AI versus human game benchmarks, *Proceedings of the International Conference on the Foundations of Digital Games*, pp. 1–8.

Cao, L. (2022). Beyond i.i.d: Non-IID thinking, informatics, and learning, *IEEE Intelligent Systems* **37**(4): 5–17.

Carbonell, J. G. (1983). Derivational analogy and its role in problem solving, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 64–69.

Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1983). An overview of machine learning, *in* R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds), *Machine Learning*, Morgan Kaufmann, pp. 3–23.

Card, S., Moran, T. P., and Newell, A. (1986). The Model Human Processor: An engineering model of human performance, *in* K. R. Boff and J. P. Thomas (eds), *Handbook of Perception and Human Performance*, Vol. 2, John Wiley & Sons, pp. 1–35.

Card, S. K. (1981). The Model Human Processor: A model for making engineering calculations of human performance, *Proceedings of the Human Factors Society Annual Meeting*, Vol. 25, pp. 301–305.

Carlson, S. M., Koenig, M. A., and Harms, M. B. (2013). Theory of mind, *Wiley Interdisciplinary Reviews: Cognitive Science* **4**(4): 391–402.

Carpenter, G. A. and Grossberg, S. (1987a). A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics, and Image Processing* **37**(1): 54–115.

Carpenter, G. A. and Grossberg, S. (1987b). Neural dynamics of category learning and recognition: Attention, memory consolidation, and amnesia, *Advances in Psychology*, Vol. 42, Elsevier, pp. 239–286.

Carpenter, G. A. and Grossberg, S. (1993). Integrating symbolic and neural processing in a self-organizing architecture for pattern recognition and prediction, *in* V. Hovanar and L. Uhr (eds), *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*, Academic Press, pp. 387–421.

Carpenter, G. A., Grossberg, S., and Reynolds, J. H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network, *Neural Networks* **4**(5): 565–588.

Carpenter, G. A., Gopal, S., Shock, B. M., and Woodcock, C. E. (2001). A neural network method for land use change classification, with application to the Nile river delta, *Technical Report TR-2001-010*, Boston University.

Carpenter, P. A., Just, M. A., and Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test, *Psychological Review* **97**(3): 404–431.

Carroll, J. B. (1993). *Human Cognitive Abilities: A Survey of Factor-Analytic Studies*, Cambridge University Press.

Carroll, J. B. (2005). The three-stratum theory of cognitive abilities, *in* D. P. Flanagan and P. L. Harrison (eds), *Contemporary Intellectual Assessment: Theories, Tests, and Issues*, The Guilford Press, pp. 69–76.

Cassimatis, N., Bignoli, P., Bugajska, M., Dugas, S., Kurup, U., Murugesan, A., and Bello, P. (2009). An architecture for adaptive algorithmic hybrids, *IEEE Transactions on Systems, Man, and Cybernetics—Part B* **40**(3): 903–914.

Cassimatis, N. L. (2005). Integrating cognitive models based on different computational methods, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 402–407.

Cassimatis, N. L. and Bignoli, P. (2011). Microcosms for testing common sense reasoning abilities, *Journal of Experimental & Theoretical Artificial Intelligence* **23**(3): 279–298.

Cassimatis, N. L., Trafton, J. G., Bugajska, M. D., and Schultz, A. C. (2004). Integrating cognition, perception and action through mental simulation in robots, *Robotics and Autonomous Systems* **49**(1-2): 13–23.

Cattell, R. and Parker, A. (2012). Challenges for brain emulation: Why is building a brain so difficult, *Natural Intelligence* **1**(3): 17–31.

Cattell, R. B. (1963). Theory of fluid and crystallized intelligence: A critical experiment, *Journal of Educational Psychology* **54**(1): 1–22.

Cavanagh, P. (2011). Visual cognition, *Vision Research* **51**(13): 1538–1551.

Cercone, N. and McCalla, G. (2012). What is knowledge representation?, *in* N. Cercone and G. McCalla (eds), *The Knowledge Frontier: Essays in the Representation of Knowledge*, Springer Science & Business Media, pp. 1–43.

Cerulo, K. A. (2009). Nonhumans in social interaction, *Annual Review of Sociology* **35**: 531–552.

Chai, W. J., Abd Hamid, A. I., and Abdullah, J. M. (2018). Working memory from the psychological and neurosciences perspectives: A review, *Frontiers in Psychology* **9**: 327922.

Chalmers, D. J., French, R. M., and Hofstadter, D. R. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology, *Journal of Experimental & Theoretical Artificial Intelligence* **4**(3): 185–211.

Chapman, D. (1987). Planning for conjunctive goals, *Artificial Intelligence* **32**(3): 333–377.

Chatham, C. H., Herd, S. A., Brant, A. M., Hazy, T. E., Miyake, A., O'Reilly, R., and Friedman, N. P. (2011). From an executive network to executive control: A computational model of the n-back task, *Journal of Cognitive Neuroscience* **23**(11): 3598–3619.

Chatterjee, S. and Zielinski, P. (2022). On the generalization mystery in deep learning, *arXiv:2203.10036*.

Chauvin, Y. and Rumelhart, D. E. (eds) (1995). *Backpropagation: Theory, Architectures, and Applications*, Lawrence Erlbaum Associates.

Chen, Y., McKinstry, J. L., and Edelman, G. M. (2013). Versatile networks of simulated spiking neurons displaying winner-take-all behavior, *Frontiers in Computational Neuroscience* **7**: 16.

Chen, Z. (1991). Analogy, systems, and intelligence, *Cybernetics and Systems: An International Journal* **22**(6): 611–616.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears, *The Journal of the Acoustical Society of America* **25**(5): 975–979.

Chikhaoui, B., Pigot, H., Beaudoin, M., Pratte, G., Bellefeuille, P., and Laudares, F. (2009). Learning a song: An ACT-R model, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Choi, D. (2009). Concurrent execution in a cognitive architecture, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 2753–2758.

Choi, D. (2023). On using generative models in a cognitive architecture for embodied agents, *Proceedings of the AAAI Symposium Series*, pp. 253–255.

Choi, D. and Langley, P. (2018). Evolution of the ICARUS cognitive architecture, *Cognitive Systems Research* **48**: 25–38.

Chollet, F. (2018). *Deep Learning with Python*, Simon and Schuster.

Chomsky, N. (1956). Three models for the description of language, *IRE Transactions on Information Theory* **2**(3): 113–124.

Chomsky, N. (1959). A review of B. F. Skinner's "Verbal Behavior", *Language* **35**(1): 26–58.

Chong, H.-Q., Tan, A.-H., and Ng, G.-W. (2007). Integrated cognitive architectures: A survey, *Artificial Intelligence Review* **28**(2): 103–130.

Chong, R. (2004). Architectural explorations for modeling procedural skill decay, *Proceedings of the International Conference on Cognitive Modeling*.

Christiansen, M. H. and Chater, N. (1993). Symbol grounding—the emperor's new theory of meaning, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 155–160.

Chuderski, A., Stettner, Z., and Orzechowski, J. (2006). Modeling individual differences in working memory search task, *Proceedings of International Conference on Cognitive Modeling*, pp. 74–79.

Cichy, R. M. and Kaiser, D. (2019). Deep neural networks as scientific models, *Trends in Cognitive Sciences* **23**(4): 305–317.

Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). High-performance neural networks for visual object classification, *Technical Report IDSIA-01-11*, Dalle Molle Institute for Artificial Intelligence.

Cochran, R. E., Lee, F. J., and Chown, E. (2006). Modeling emotion: Arousal's impact on memory, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1133–1138.

Cohen, L. J. (1981). Can human irrationality be experimentally demonstrated?, *Behavioral and Brain Sciences* **4**(3): 317–331.

Cohen, M. A. and Grossberg, S. (1986). Neural dynamics of speech and language coding: Developmental programs, perceptual grouping, and competition for short-term memory, *Human Neurobiology* **5**(1): 1–22.

Cohen, P. R. (2005). If not Turing's test, then what?, *AI Magazine* **26**(4): 61–67.

Colins, A. M. and Loftus, E. F. (1975). A spreading-activation theory of semantic memory, *Psychological Review* **82**: 407–428.

Companion, M. A. and Corso, G. M. (1982). Task taxonomies: A general review and evaluation, *International Journal of Man-Machine Studies* **17**(4): 459–472.

Connell, J. H. (1989). A colony architecture for an artificial creature, *Technical Report 1151*, MIT.

Connolly, K. and Prettyman, A. (2024). Perceptual Learning, *in* E. N. Zalta and U. Nodelman (eds),

The Stanford Encyclopedia of Philosophy, Fall 2024 edn, Metaphysics Research Lab, Stanford University.

Contreras Kallens, P., Dale, R., and Christiansen, M. H. (2022). Quantifying interdisciplinarity in cognitive science and beyond, *Topics in Cognitive Science* **14**(3): 634–645.

Conway, A. R., Kane, M. J., Bunting, M. F., Hambrick, D. Z., Wilhelm, O., and Engle, R. W. (2005). Working memory span tasks: A methodological review and user's guide, *Psychonomic Bulletin & Review* **12**: 769–786.

Conway, M. A. (2008). Exploring episodic memory, *in* E. Dere, A. Easton, L. Nadel, and J. P. Huston (eds), *Handbook of Behavioral Neuroscience*, Vol. 18, Elsevier, pp. 19–29.

Cook, J. (2023). Branding babble: What happened when ChatGPT took the Turing test, `https://tinyurl.com/chatgpt-marketing`.

Coombs, D., Herman, M., Hong, T.-H., and Nashman, M. (1998). Real-time obstacle avoidance using central flow divergence, and peripheral flow, *IEEE Transactions on Robotics and Automation* **14**(1): 49–59.

Coombs, D., Murphy, K., Lacaze, A., and Legowik, S. (2000). Driving autonomously off-road up to 35 km/h, *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 186–191.

Cooper, R., Fox, J., Farringdon, J., and Shallice, T. (1996). A systematic methodology for cognitive modelling, *Artificial Intelligence* **85**(1-2): 3–44.

Cooper, R. P. (2006). Cognitive architectures as Lakatosian research programs: Two case studies, *Philosophical Psychology* **19**(2): 199–220.

Cooper, R. P. and Guest, O. (2014). Implementations are not specifications: Specification, replication and experimentation in computational cognitive modeling, *Cognitive Systems Research* **27**: 42–49.

Cooper, R. P. and Peebles, D. (2015). Beyond single-level accounts: The role of cognitive architectures in cognitive scientific explanation, *Topics in Cognitive Science* **7**(2): 243–258.

Copeland, B. J. (2000). The Turing test, *Minds and Machines* **10**(4): 519–539.

Copeland, J. B. (ed.) (2004). *The Essential Turing*, Oxford University Press.

Corker, K. and Smith, B. (1993). An architecture and model for cognitive engineering simulation analysis: Application to advanced aviation automation, *Proceedings of the AIAA Computing in Aerospace Conference*, pp. 1–8.

Corker, K., Pisanich, G., and Bunzo, M. (1997). A cognitive system model for human/automation dynamics in airspace management, *Proceedings of the European/US Symposium on Air Traffic Management*.

Corker, K. M. (1999). Human performance simulation in the analysis of advanced air traffic management, *Proceedings of the IEEE Winter Simulation Conference*, pp. 821–828.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity, *Behavioral and Brain Sciences* **24**(1): 87–114.

Cowan, N. (2008). Sensory memory, *in* J. H. Byrne (ed.), *Learning and Memory: A Comprehensive Reference*, Academic Press, pp. 23–32.

Cox, M., Alavi, Z., Dannenhauer, D., Eyorokon, V., Munoz-Avila, H., and Perlis, D. (2016). MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3712–3718.

Cox, M., Mohammad, Z., Kondrakunta, S., Gogineni, V. R., Dannenhauer, D., and Larue, O. (2022). Computational metacognition, *Proceedings of the Annual Conference on Advances in Cognitive Systems*.

Cox, M. T. (2005). Metacognition in computation: A selected research review, *Artificial Intelligence* **169**(2): 104–141.

Cox, M. T. and Ram, A. (1994). Interacting learning-goals: Treating learning as a planning task, *European Workshop on Advances in Case-Based Reasoning*, Springer, pp. 60–74.

Craig, I. D. (1988). Blackboard systems, *Artificial Intelligence Review* **2**(2): 103–118.

Craik, K. J. W. (1943). *The Nature of Explanation*, Cambridge University Press.

Crawford, E., Gingerich, M., and Eliasmith, C. (2016). Biologically plausible, human-scale knowledge representation, *Cognitive Science* **40**(4): 782–821.

Crick, F. (1989). The recent excitement about neural networks, *Nature* **337**(6203): 129–132.

Crossman, J., Wray, R. E., Jones, R. M., and Lebiere, C. (2004a). A high level symbolic representation for behavior modeling, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*, pp. 212–220.

Crossman, J., Wray, R., Nielsen, P., Jones, R. M., Wallace, A., and Lebiere, C. (2004b). A high-level symbolic representation for intelligent agents across multiple architectures, *Technical Report AFRL-HE-WP-TR-2005-0006*, United States Air Force Research Laboratory.

da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., and dos Reis Alves, S. F. (2017). Introduction, *Artificial Neural Networks: A Practical Course*, Springer, pp. 3–19.

Daily, L. Z., Lovett, M. C., and Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account, *Cognitive Science* **25**(3): 315–353.

Damasio, A. (1994). *Descartes' Error: Emotion, Reason and the Human Brain*, Avon Books.

Damassino, N. (2020). The Questioning Turing Test, *Minds and Machines* **30**(4): 563–587.

DARPA (2005). Biologically-inspired cognitive architectures (BICA) proposer information pamphlet, `http://www.darpa.mil/ipto/solicitations/open/05-18_PIP.htm`. Accessible via `http://web.archive.org`.

Davis, R. and King, J. (1975). An overview of production systems, *Technical Report STAN-CS-75-524*, Stanford University.

Davis, R. and King, J. J. (1984). The origin of rule-based systems in AI, *in* B. G. Buchanan and E. H. Shortliffe (eds), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley.

Davis, R., Shrobe, H., and Szolovits, P. (1993). What is a knowledge representation?, *AI Magazine* **14**(1): 17–17.

Dawkins, R. (1976). Hierarchical organisation: A candidate principle for ethology, *in* P. P. G. Batson and R. A. Hinde (eds), *Growing Points in Ethology*, Cambridge University Press, pp. 7–54.

Day, B. L. and Fitzpatrick, R. C. (2005). The vestibular system, *Current Biology* **15**(15): R583–R586.

De Houwer, J., Barnes-Holmes, D., and Moors, A. (2013). What is learning? On the nature and merits of a functional definition of learning, *Psychonomic Bulletin & Review* **20**(4): 631–642.

De Silva, L., Meneguzzi, F., and Logan, B. (2020). BDI agent architectures: A survey, *Proceedings of the International Joint Conferences on Artificial Intelligence*, pp. 4914–4921.

Dean, T. L. and Boddy, M. S. (1988). An analysis of time-dependent planning, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 88, pp. 49–54.

Dehghani, M., Tay, Y., Gritsenko, A. A., Zhao, Z., Houlsby, N., Diaz, F., Metzler, D., and Vinyals, O. (2021). The benchmark lottery, *arXiv:2107.07002*.

DeJong, G. (1988). An introduction to explanation-based learning, *in* H. E. Shrobe (ed.), *Exploring Artificial Intelligence*, Morgan Kaufmann, pp. 45–81.

DeJong, G. and Mooney, R. (1986). Explanation-based learning: An alternative view, *Machine Learning* **1**(2): 145–176.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255.

Deutsch, S. and Cramer, N. (1998). OMAR human performance modeling in a decision support experiment, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 42, pp. 1232–1236.

Deutsch, S. and Pew, R. (2003). Modeling the NASA baseline and SVS-equipped approach and landing scenarios in D-OMAR, *Proceedings of the Conference Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*, pp. 143–164.

Deutsch, S. and Pew, R. (2019). Modeling human error in a real-world teamwork environment, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 274–279.

Deutsch, S., Pew, R., Granville, R., Roberts, B., and Mulvehill, A. (1998). Automating maintenance instructions study: Procedure planning technologies, *Technical Report BBN 8231*, BBN Systems and Technologies.

Deutsch, S., MacMillan, J., Cramer, N. L., and Chopra, S. (2014). Operability model architecture demonstration final report, *Technical Report AL/HR-TR-1996-0161*, BBN Corporation.

Devlin, S., Georgescu, R., Momennejad, I., Rzepecki, J., Zuniga, E., Costello, G., Leroy, G., Shaw, A., and Hofmann, K. (2021). Navigation Turing Test (NTT): Learning to evaluate human-like navigation, *Proceedings of the International Conference on Machine Learning*, pp. 2644–2653.

Dickison, D. and Taatgen, N. A. (2007). ACT-R models of cognitive control in the abstract decision making task, *Proceedings of the International Conference on Cognitive Modeling*, pp. 79–84.

Dickmanns, E. D. (1990). Visual dynamic scene understanding exploiting high-level spatio-temporal models, *Proceedings of the International Conference on Pattern Recognition*, Vol. 2, IEEE, pp. 373–378.

Dinsmore, J. (1992). Thunder in the gap, *in* J. Dinsmore (ed.), *The Symbolic and Connectionist Paradigms*, Psychology Press, pp. 1–24.

d'Inverno, M., Kinny, D., Luck, M., and Wooldridge, M. (1998). A formal specification of dMARS, *Proceedings of the International Workshop on Agent Theories, Architectures, and Languages*, pp. 155–176.

D'Mello, S. K., Ramamurthy, U., Negatu, A., and Franklin, S. (2006). A procedural learning mechanism for novel skill acquisition, *in* T. Kovacs and A. R. Marshall (eds), *Proceeding of AISB'06: Adaptation in Artificial and Biological Systems*, Vol. 184–185, University of Bristol.

Dobrev, D. (2005). Formal definition of artificial intelligence, *International Journal ITA* **12**(3): 277–285.

Doerig, A. et al. (2023). The neuroconnectionist research programme, *Nature Reviews Neuroscience* **24**(7): 431–450.

Dong, G., Zhao, J., Hui, T., Guo, D., Wang, W., Feng, B., Qiu, Y., Gongque, Z., He, K., Wang, Z., and Xu, W. (2023). Revisit input perturbation problems for LLMs: A unified robustness evaluation framework for noisy slot filling task, *CCF International Conference on Natural Language Processing and Chinese Computing*, Springer, pp. 682–694.

Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D. (1999). Adjustable autonomy for human-centered autonomous systems, *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems*, pp. 16–35.

Douglass, S., Ball, J., and Rodgers, S. (2009). Large declarative memories in ACT-R, *Proceedings of the International Conference of Cognitive Modeling*.

Doumas, L., Morrison, R., and Richland, L. (2010). Differences in the development of analogy across cultures: A computational account, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Dowe, D. L. and Hernández-Orallo, J. (2014). How universal can an intelligence test be?, *Adaptive Behavior* **22**(1): 51–69.

Downey, R. G. and Fellows, M. R. (1995). Fixed-parameter tractability and completeness i: Basic results, *SIAM Journal on computing* **24**(4): 873–921.

Dreyfus, S. E. (1990). Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure, *Journal of Guidance, Control, and Dynamics* **13**(5): 926–928.

Drummond, M., Bresina, J., and Kedar, S. (1991). The Entropy Reduction Engine: Integrating planning, scheduling, and control, *ACM SIGART Bulletin* **2**(4): 61–65.

Duan, J., Yu, S., Tan, H. L., Zhu, H., and Tan, C. (2022). A survey of embodied AI: From simulators to research tasks, *IEEE Transactions on Emerging Topics in Computational Intelligence* **6**(2): 230–244.

Duch, W., Oentaryo, R. J., and Pasquier, M. (2008). Cognitive architectures: Where do we go from here?, *Proceedings of the Artificial General Intelligence Conference*, Vol. 171, pp. 122–136.

Duff, A., Rennó-Costa, C., Marcos, E., Luvizotto, A. L., Giovannucci, A., Sanchez-Fibla, M., Bernardet, U., and Verschure, P. F. (2010). Distributed adaptive control: A proposal on the neuronal organization of adaptive goal oriented behavior, *in* O. Sigaud and J. Peters (eds), *From Motor Learning to Interaction Learning in Robots*, Springer, pp. 15–41.

Dupoux, E. (2018). Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner, *Cognition* **173**: 43–59.

Duro, R., Bellas, F., and Becerra, J. (2010). Evolutionary architecture for lifelong learning and real-time operation in autonomous robots, *in* P. Angelov, D. P. Filev, and N. Kasabov (eds), *Evolving Intelligent Systems: Methodology and Applications*, Wiley Online Library, pp. 365–400.

Dutson, M., Li, Y., and Gupta, M. (2023). Spike-based anytime perception, *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 5294–5304.

Dybala, T., Tecuci, G., and Rezazad, H. (1996). The shared expertise model for teaching interactive design assistants, *Engineering Applications of Artificial Intelligence* **9**(6): 611–626.

Edelman, G. M. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection*, Basic Books.

Edelman, G. M. (2007). Learning in and from brain-based devices, *Science* **318**(5853): 1103–1105.

Eggleston, R. G., Young, M. J., and McCreight, K. L. (2000). Distributed cognition: A new type of human performance model, *Proceedings of the AAAI Fall Symposium*, pp. 8–14.

Eliasmith, C. (2013). *How to Build a Arain: A Neural Architecture for Biological Cognition*, Oxford University Press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., and Rasmussen, D. (2012). A large-scale model of the functioning brain, *Science* **338**(6111): 1202–1205.

Eliasmith, C., Gosmann, J., and Choo, X. (2016). BioSpaun: A large-scale behaving brain model with complex neurons, *arXiv:1602.05220*.

Elkins, L., Sellers, D., and Monach, W. R. (2010). The Autonomous Maritime Navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles, *Journal of Field Robotics* **27**(6): 790–818.

Ellsworth, P. C. and Scherer, K. R. (2009). Appraisal processes in emotion, *in* R. J. Davidson, K. R. Sherer, and H. H. Goldsmith (eds), *Handbook of Affective Sciences*, Oxford University Press, pp. 572–595.

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness: A Connectionist Perspective on Development*, MIT Press.

Enholm, I. M., Papagiannidis, E., Mikalef, P., and Krogstie, J. (2022). Artificial intelligence and business value: A literature review, *Information Systems Frontiers* **24**(5): 1709–1734.

Epstein, R. (1992a). The quest for the thinking computer, *AI Magazine* **13**(2): 81–95.

Epstein, S. L. (1992b). The role of memory and concepts in learning, *Minds and Machines* **2**(3): 239–265.

Epstein, S. L. (1994). Toward an ideal trainer, *Machine Learning* **15**(3): 251–277.

Epstein, S. L. (1997). Representation and reasoning for pragmatic navigation, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19–28.

Epstein, S. L. and Petrovic, S. (2008). Learning expertise with bounded rationality and self-awareness, *in* Y. Hamadi and E. M. F. Saubion (eds), *AAAI Workshop on Metareasoning: Thinking about thinking*, pp. 40–47.

Epstein, S. L., Freuder, E. C., Wallace, R., Morozov, A., and Samuels, B. (2002). The adaptive constraint engine, *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, pp. 525–540.

Epstein, S. L., Gordon, J., Passonneau, R., and Ligorio, T. (2010). Toward spoken dialogue as mutual agreement, *Proceedings of the AAAI Conference on Metacognition for Robust Social Systems*, pp. 14–21.

Epstein, S. L., Passonneau, R., Gordon, J., and Ligorio, T. (2011). The role of knowledge and certainty in understanding for dialogue, *Proceedings of the AAAI Fall Symposium*, pp. 90–97.

Epstein, S. L., Schneider, E., Ozgelen, A. T., Munoz, J. P., Costantino, M., Sklar, E. I., and Parsons, S. (2012). Applying FORR to human/multi-robot teams, *Proceedings of the International Conference on Human-Robot Interaction, Workshop on Human-Agent-Robot Teams*.

Ericsson, K. A. and Kintsch, W. (1995). Long-term working memory, *Psychological Review* **102**(2): 211.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R. (1980). The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty, *Computing Surveys* **12**(2): 213–253.

Escobedo, R., Smith, S. D., and Caudell, T. P. (1993). A neural information retrieval system, *International Journal of Advanced Manufacturing Technology* **8**(4): 269–273.

Estes, W. K. (1991). Cognitive architectures from the standpoint of an experimental psychologist, *Annual Review of Psychology* **42**: 1–28.

Evertsz, R., Ritter, F. E., Russell, S., and Shepherdson, D. (2007). Modeling rules of engagement in computer generated forces, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*, pp. 613–625.

Faghihi, U., McCall, R., and Franklin, S. (2012). A computational model of attentional learning in a cognitive agent, *Biologically Inspired Cognitive Architectures* **2**: 25–36.

Fanelli, D. (2012). Negative results are disappearing from most disciplines and countries, *Scientometrics* **90**(3): 891–904.

Feigenbaum, E. A. (2003). Some challenges and grand challenges for computational intelligence, *Journal of the ACM* **50**(1): 32–40.

Feldman, J. A. and Ballard, D. H. (1982). Connectionist models and their properties, *Cognitive Science* **6**(3): 205–254.

Feldman, V. and Kokinov, B. (2009). Anxiety restricts the analogical search in an analogy generation task, *in* B. Kokinov, K. Holyoak, and D. Gentner (eds), *New Frontiers in Analogy Research*, NBU Press, pp. 117–126.

Ferreira, W. P., Maria do Carmo, G. S., Lotufo, A. P., and Minussi, C. R. (2006). Transient stability analysis of electric energy systems via a fuzzy ART-ARTMAP neural network, *Electric Power Systems Research* **76**(6-7): 466–475.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefer, N., and Welty, C. (2010). Building Watson: An overview of the DeepQA project, *AI Magazine* **31**(3): 59–79.

Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving, *Artificial Intelligence* **2**(3–4): 189–208.

Fink, E. and Blythe, J. (1998). A complete bidirectional planner, *Proceedings of the International Conference on AI Planning and Scheduling*, pp. 78–85.

Fink, J. (2013). What is (correct) practical reasoning?, *Acta Analytica* **28**(4): 471–482.

Fink, J. and Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the World Wide Web, *User Modeling and User-Adapted Interaction* **10**(2): 209–249.

Firby, R. J. (1989). *Adaptive Execution in Complex Dynamic Worlds*, PhD thesis, Yale University.

Firby, R. J., Kahn, R. E., Prokopowicz, P. N., and Swain, M. J. (1995). An architecture for vision and action, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 72–79.

Firestone, C. and Scholl, B. J. (2016). Cognition does not affect perception: Evaluating the evidence for "top-down" effects, *Behavioral and Brain Sciences* **39**: e229.

Flanagan, D. P. and Dixon, S. G. (2014). The Cattell-Horn-Carroll theory of cognitive abilities, *in* C. R. Reynolds, K. J. Vannest, and E. Fletcher-Janzen (eds), *Encyclopedia of Special Education*, John Wiley & Sons.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry, *American Psychologist* **34**(10): 906–911.

Fleischer, J. G. and Edelman, G. M. (2009). Brain-Based Devices. An embodied approach to linking nervous system structure and function to behavior, *IEEE Robotics & Automation Magazine* **16**(3): 33–41.

Fleischer, J. G., Gally, J. A., Edelman, G. M., and Krichmar, J. L. (2007). Retrospective and prospective responses arising in a modeled hippocampus during maze navigation by a brain-based device, *Proceedings of the National Academy of Sciences* **104**(9): 3556–3561.

Fletcher, L. and Carruthers, P. (2012). Metacognition and reasoning, *Philosophical Transactions of the Royal Society B: Biological Sciences* **367**(1594): 1366–1378.

Flynn, A. M. and Brooks, R. A. (1989). Battling reality, *Technical Report 1148*, MIT.

Forbus, K. D. (2010). AI and cognitive science: The past and next 30 years, *Topics in Cognitive Science* **2**(3): 345–356.

Forbus, K. D. (2012). How minds will be built, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, Cognitive Systems Foundation, pp. 47–58.

Forbus, K. D. and Hinrichs, T. R. (2006). Companion cognitive systems: A step toward human-level AI, *AI Magazine* **27**(2): 83–83.

Forbus, K. D., Gentner, D., and Law, K. (1995). MAC/FAC: A model of similarity-based retrieval, *Cognitive Science* **19**(2): 141–205.

Forbus, K. D., Liang, C., and Rabkina, I. (2017). Representation and computation in cognitive models, *Topics in Cognitive Science* **9**(3): 694–718.

Forbus, K., Usher, J., Lovett, A., Lockwood, K., and Wetzel, J. (2011). CogSketch: Sketch understanding for cognitive science research and for education, *Topics in Cognitive Science* **3**(4): 648–666.

Francis, G. (2012). The psychology of replication and replication in psychology, *Perspectives on Psychological Science* **7**(6): 585–594.

Frankish, K. and Ramsey, W. (eds) (2012). *The Cambridge Handbook of Cognitive Science*, Cambridge University Press.

Franklin, S. (2003). An autonomous software agent for Navy personnel work: A case study, *Proceedings of the AAAI Spring Symposium*, pp. 60–65.

Franklin, S. and Baars, B. J. (2010). Spontaneous remembering is the norm: What integrative models tell us about human consciousness and memory, *in* J. H. Mace (ed.), *The Act of Remembering: Toward an Understanding of How We Recall the Past*, Blackwell Publishing Ltd, pp. 83–110.

Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents, *International Workshop on Agent Theories, Architectures, and Languages*, pp. 21–35.

Franklin, S. and Graesser, A. (1999). A software agent model of consciousness, *Consciousness and Cognition* **8**(3): 285–301.

Franklin, S. and Patterson Jr, F. (2006). The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent, *Proceedings of the International Conference on Integrated Design and Process Technology*, pp. 764–1004.

Franklin, S. and Ramamurthy, U. (2006). Motivations, values and emotions: Three sides of the same coin, *Proceedings of the International Workshop on Epigenetic Robotics*, pp. 41–48.

Franklin, S., D'Mello, S., Baars, B. J., and Ramamurthy, U. (2009). Evolutionary pressures for perceptual stability and self as guides to machine consciousness, *International Journal of Machine Consciousness* **1**(1): 99–110.

Franklin, S., Strain, S., Snaider, J., McCall, R., and Faghihi, U. (2012). Global workspace theory, its LIDA model and the underlying neuroscience, *Biologically Inspired Cognitive Architectures* **1**: 32–43.

Franklin, S., Madl, T., D'mello, S., and Snaider, J. (2013). LIDA: A systems-level architecture for cognition, emotion, and learning, *IEEE Transactions on Autonomous Mental Development* **6**(1): 19–41.

Franklin, S., Madl, T., Strain, S., Faghihi, U., Dong, D., Kugele, S., Snaider, J., Agrawal, P., and Chen, S. (2016). A LIDA cognitive model tutorial, *Biologically Inspired Cognitive Architectures* **16**: 105–130.

Freed, M. (1998). Managing multiple tasks in complex, dynamic environments, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 921–927.

Freed, M. and Remington, R. (1999). A conceptual framework for predicting error in complex human-machine environments, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 356–361.

Freed, M. and Remington, R. W. (1997). Managing decision resources in plan execution, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 322–328.

Freeman, J. (2015). Open source tools for large-scale neuroscience, *Current Opinion in Neurobiology* **32**: 156–163.

French, R. M. (1999). Catastrophic forgetting in connectionist networks, *Trends in Cognitive Sciences* **3**(4): 128–135.

French, R. M. (2002). The computational modeling of analogy-making, *Trends in Cognitive Sciences* **6**(5): 200–205.

Friedman, S. and Forbus, K. (2011). Repairing incorrect knowledge with model formulation and metareasoning, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 887–893.

Friedman, S., Taylor, J., and Forbus, K. (2009). Learning naïve physics models by analogical generalization, *Proceedings of the International Analogy Conference*, pp. 168–177.

Frijda, N. H. (1994). Emotions require cognitions, even if simple ones, *in* P. Ekman and R. Davison (eds), *The Nature of Emotions: Fundamental Questions*, Oxford University Press, pp. 197–202.

Froese, V. and Hertrich, C. (2023). Training neural networks is np-hard in fixed dimension, *Advances in Neural Information Processing Systems*, pp. 44039–44049.

Fu, W.-T. and Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model, *Journal of Experimental Psychology: General* **135**(2): 184.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics* **36**(4): 193–202.

Fulkerson, M. (2014). Rethinking the senses and their interactions: The case for sensory pluralism, *Frontiers in Psychology* **5**: 1426.

Fum, D. and Stocco, A. (2004). Memory, emotion, and rationality: An ACT-R interpretation for gambling task results, *Proceedings of the International Conference on Cognitive Modeling (ICCM)*, pp. 106–111.

Furber, S. and Temple, S. (2007). Neural systems engineering, *Journal of the Royal Society Interface* **4**(13): 193–206.

Fuster, J. (2017). Prefrontal cortex in decision-making: The perception–action cycle, *in* J.-C. Dreher and L. Tremblay (eds), *Decision Neuroscience: An Integrative Approach*, Elsevier, pp. 95–105.

Galotti, K. M. (1989). Approaches to studying formal and everyday reasoning, *Psychological Bulletin* **105**(3): 331.

Gamrian, S. and Goldberg, Y. (2019). Transfer learning for related reinforcement learning tasks via image-to-image translation, *Proceedings of the International Conference on Machine Learning*, pp. 2063–2072.

Gao, L. et al. (2020). The Pile: An 800GB dataset of diverse text for language modeling, *arXiv:2101.00027*.

Garcia-Marques, L. and Ferreira, M. B. (2011). Friends and foes of theory construction in psychological science: Vague dichotomies, unified theories of cognition, and the new experimentalism, *Perspectives on Psychological Science* **6**(2): 192–201.

Gat, E. (1991a). Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, *ACM SIGART Bulletin* **2**(4): 70–74.

Gat, E. (1991b). *Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots*, PhD thesis, Virginia Polytechnic Institute and State University.

Gat, E. (1993). On the role of stored internal state in the control of autonomous mobile robots, *AI Magazine* **14**(1): 64–73.

Gat, E. (1998). On three-layer architectures, *in* D. Kortenkamp, P. R. Bonasso, and R. Murphy (eds), *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, AAAI Press, pp. 195–210.

Gawron, V. J. et al. (1991). Human factors taxonomy, *Proceedings of the Human Factors Society Annual Meeting*, pp. 1284–1287.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks, *Advances in Neural Information Processing Systems* **31**: 7538–7550.

Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W. (2021). Partial success in closing the gap between human and machine vision, *Advances in Neural Information Processing Systems* **34**: 23885–23899.

Geman, D., Geman, S., Hallonquist, N., and Younes, L. (2015). Visual Turing test for computer vision systems, *Proceedings of the National Academy of Sciences* **112**(12): 3618–3623.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy, *Cognitive Science* **7**(2): 155–170.

Gentner, D. (2003). Analogical reasoning, psychology of, *Encyclopedia of cognitive science* **1**: 106–112.

Gentner, D. (2010). Psychology in cognitive science: 1978–2038, *Topics in Cognitive Science* **2**(3): 328–344.

Gentner, D. and Forbus, K. D. (2011). Computational models of analogy, *Wiley Interdisciplinary Reviews: Cognitive Science* **2**(3): 266–276.

Georgeff, M., Pell, B., Pollack, M., Tambe, M., and Wooldridge, M. (1999). The belief-desire-intention model of agency, *International Workshop on Agent Theories, Architectures, and Languages*.

Georgeff, M. P. and Ingrand, F. F. (1989). Monitoring and control of spacecraft systems using procedural reasoning, *Proceedings of the NASA Annual Workshop on Space Operations Automation and Robotics*, pp. 209–217.

Georgeff, M. P. and Lansky, A. L. (1987). Reactive reasoning and planning, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 677–682.

Gevarter, W. B. (1982). An overview of expert systems, *Technical Report NBSIR 82-2505*, US Department of Commerce.

Gever, D., Daley, M., Babina, M., Snihurowych, B., Bonacci, L., and Bolkhovsky, J. (2020). Integration of a computational auditory scene analysis software architecture with the ARCADIA cognitive model: Software design, usage, implementation, demonstration, *Technical Report NSMRL/F1703/TR–2020-1340*, Naval Submarine Medical Research Laboratory.

Ghirlanda, S. and Enquist, M. (2003). A century of generalization, *Animal Behaviour* **66**(1): 15–36.

Gibney, E. (2022). Is AI fuelling a reproducibility crisis in science, *Nature* **608**: 250–251.

Gil, L., Barat, J. M., Escriche, I., Garcia-Breijo, E., Martínez-Máñez, R., and Soto, J. (2008). An electronic tongue for fish freshness analysis using a thick-film array of electrodes, *Microchimica Acta* **163**(1-2): 121–129.

Gill, T. G. (1995). Early expert systems: Where are they now?, *MIS Quarterly* **19**(1): 51–81.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning, *International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 80–89.

Ginsberg, M. L. (1989). Universal planning: An (almost) universally bad idea, *AI Magazine* **10**(4): 40–40.

Gluck, K. A. and Pew, R. W. (eds) (2006). *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation*, Psychology Press.

Gobet, F. (1998). Memory for the meaningless: How chunks help, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 398–403.

Gobet, F. (2000). Some shortcomings of long-term working memory, *British Journal of Psychology* **91**(4): 551–570.

Gobet, F. (2013). Chunks and templates in semantic long-term memory, *in* J. J. Stazewski (ed.), *Expertise and Skill Acquisition*, Psychology Press, pp. 117–145.

Gobet, F. and Lane, P. (2012). Chunking mechanisms and learning, *in* N. M. Seel (ed.), *Encyclopedia of the Sciences of Learning*, Springer, pp. 541–544.

Goertzel, B. (2010). Toward a formal characterization of real-world general intelligence, *Proceedings of the Conference on Artificial General Intelligence*, pp. 74–79.

Goertzel, B. (2012a). Modifying the DeSTIN perception architecture to enable representationally transparent deep learning, `https://goertzel.org/papers/Uniform_DeSTIN_paper_v2.pdf`.

Goertzel, B. (2012b). Perception processing for general intelligence: Bridging the symbolic/subsymbolic gap, *Proceedings of the International Conference on Artificial General Intelligence*, Springer.

Goertzel, B. and Duong, D. (2009). OpenCog NS: A deeply-interactive hybrid neural-symbolic cognitive architecture designed for global/local memory synergy, *Proceedings of the AAAI Fall Symposium*, pp. 63–68.

Goertzel, B. and Yu, G. (2014). From here to AGI: A roadmap to the realization of human-level artificial general intelligence, *Proceedings of the International Joint Conference on Neural Networks*, pp. 1525–1533.

Goertzel, B., de Garis, H., Pennachin, C., Geisweiller, N., Araujo, S., Pitt, J., Chen, S., Lian, R., Jiang, M., Yang, Y., and Huang, D. (2010a). OpenCogBot: Achieving generally intelligent virtual agent control and humanoid robotics via cognitive synergy, *Proceedings of International Conference on Artificial Ingelligence*, pp. 1–12.

Goertzel, B., Lian, R., Arel, I., De Garis, H., and Chen, S. (2010b). A world survey of artificial brain projects. Part II: Biologically inspired cognitive architectures, *Neurocomputing* **74**(1–3): 30–49.

Goertzel, B., Ke, S., Lian, R., O'Neill, J., Sadeghi, K., Wang, D., Watkins, O., and Yu, G. (2013). The CogPrime architecture for embodied artificial general intelligence, *IEEE Symposium on Computational Intelligence for Human-Like Intelligence*, pp. 60–67.

Goertzel, B., Pennachin, C., and Geisweiller, N. (2014). Brief survey of cognitive architectures, *Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy*, Atlantis Press, pp. 101–142.

Goldinger, S. D. and Azuma, T. (2003). Puzzle-solving science: The quixotic quest for units in speech perception, *Journal of Phonetics* **31**(3–4): 305–320.

Goldstone, R. and Leydesdorff, L. (2006). The import and export of cognitive science, *Cognitive Science* **30**(6): 983–993.

González-Santamarta, M. Á., Rodríguez-Lera, F. J., Álvarez-Aparicio, C., Guerrero-Higueras, Á. M., and Fernández-Llamas, C. (2020). MERLIN: A cognitive architecture for service robots, *Applied Sciences*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets, *Advances in Neural Information Processing Systems* pp. 2672–2680.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*, MIT Press.

Goodman, N. (1965). *Fact, Fiction, and Forecast*, Oxford University Press.

Gore, B. F., Hooey, B. L., Wickens, C. D., and Scott-Nash, S. (2009). A computational implementation of a human attention guiding mechanism in MIDAS v5, *Proceedings of the International Conference on Digital Human Modeling*, pp. 237–246.

Gore, B. F., Hooey, B. L., and Foyle, D. C. (2011). NASA's use of human performance models for NextGen concept development and evaluations, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*.

Gosmann, J. and Eliasmith, C. (2015). A spiking neural model of the n-back task, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 812–817.

Grace, K. and Christiano, P. (2015). Brain performance in TEPS, `https://aiimpacts.org/brain-performance-in-teps/`.

Grant, D. A. (1962). Testing the null hypothesis and the strategy and tactics of investigating theoretical models, *Psychological Review* **69**(1): 54–61.

Gratch, J. and Marsella, S. (2015). Appraisal models, *in* R. Calvo, S. D'Mello, J. Gratch, and A. Kappas (eds), *The Oxford Handbook of Affective Computing*, Oxford University Press, pp. 54–67.

Gravitz, L. (2019). The importance of forgetting, *Nature* **571**: S12–S14.

Gray, E. K., Watson, D., Payne, R., and Cooper, C. (2001). Emotion, mood, and temperament: Similarities, differences, and a synthesis, *in* R. L. Payne and C. Cooper (eds), *Emotions at Work: Theory, Research and Applications for Management*, John Wiley & Sons, pp. 21–43.

Graylin, A., Kjolaas, K. A. H., Loflin, J., and Walker III, J. D. (1998). Symbolics, Inc.: A failure of heterogeneous engineering, *Technical report*, MIT.

Greff, K., Van Steenkiste, S., and Schmidhuber, J. (2020). On the binding problem in artificial neural networks, *arXiv:2012.05208*.

Griffin, D. R. (1984). Animal thinking, *American Scientist* **72**(5): 456–464.

Griffiths, T. L., Lieder, F., and Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic, *Topics in Cognitive Science* **7**(2): 217–229.

Grinberg, M. and Kokinov, B. (2019). Simulation of episode blending in the AMBR model, *Proceedings of European Cognitive Science Conference*, pp. 151–155.

Gross, C. G. (2002). Genealogy of the "grandmother cell", *Neuroscientist* **8**(5): 512–518.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance, *Cognitive Science* **11**(1): 23–63.

Grossberg, S. (2003). Resonant neural dynamics of speech perception, *Journal of Phonetics* **31**(3-4): 423–445.

Grossberg, S. (2007a). Consciousness clears the mind, *Neural Networks* **20**(9): 1040–1053.

Grossberg, S. (2007b). Towards a unified theory of neocortex: Laminar cortical circuits for vision and cognition, *Progress in Brain Research* **165**: 79–104.

Grossberg, S. (2020). A path toward explainable AI and autonomous adaptive intelligence: Deep learning, adaptive resonance, and models of perception, emotion, and action, *Frontiers in Neurorobotics* **14**: 36.

Grossberg, S. (2021). *Conscious Mind, Resonant Brain: How Each Brain Makes a Mind*, Oxford University Press.

Grossberg, S. and Merrill, J. W. (1992). A neural network model of adaptively timed reinforcement learning and hippocampal dynamics, *Cognitive Brain Research* **1**(1): 3–38.

Grossberg, S. and Myers, C. W. (2000). The resonant dynamics of speech perception: Interword integration and duration-dependent backward effects, *Psychological Review* **107**(4): 735–767.

Grossberg, S. and Seidman, D. (2006). Neural dynamics of autistic behaviors: Cognitive, emotional,

and timing substrates, *Psychological Review* **113**(3): 483.
Grossberg, S. and Versace, M. (2008). Spikes, synchrony, and attentive learning by laminar thalamocortical circuits, *Brain Research* **1218**: 278–312.
Grossberg, S., Govindarajan, K. K., Wyse, L. L., and Cohen, M. A. (2004). ARTSTREAM: A neural network model of auditory scene analysis and source segregation, *Neural Networks* **17**(4): 511–536.
Großmann, P., Siebers, M., and Schmid, U. (2012). MoralLISA—An extension of the analogy model LISA for moral decision making, *Proceedings of the KI Workshop on Human Reasoning and Automated Deduction*, pp. 9–16.
Gudwin, R., Paraense, A., de Paula, S. M., Fróes, E., Gibaut, W., Castro, E., Figueiredo, V., and Raizer, K. (2017). The multipurpose enhanced cognitive architecture (MECA), *Biologically Inspired Cognitive Architectures* **22**: 20–34.
Gudwin, R., Paraense, A., de Paula, S., Fróes, E., Gibaut, W., Castro, E., Figueiredo, V., and Raizer, K. (2018). An overview of the multipurpose enhanced cognitive architecture (MECA), *Procedia Computer Science* **123**: 155–160.
Gudwin, R., Rohmer, E., Paraense, A., Froes, E., Gibaut, W., Oliveira, I., Rocha, S., Raizer, K., and Feljan, A. V. (2020a). The TROCA project: An autonomous transportation robot controlled by a cognitive architecture, *Cognitive Systems Research* **59**: 179–197.
Gudwin, R., Rohmer, E., Paraense, A. L. O., Fróes, E., Gibaut, W., Oliveira, I., Rocha, S., Raizer, K., and Feljan, A. V. (2020b). A cognitive architecture for a transportation robotic system, *Proceedings of the Annual Meeting of the BICA Society*, pp. 110–116.
Gunetti, P., Dodd, T., and Thompson, H. (2010). A software architecture for Autonomous UAV Mission Management and Control, *AIAA InfotechAerospace*, pp. 3305–3316.
Gunzelmann, G. and Gluck, K. (2009). An integrative approach to understanding and predicting the consequences of fatigue on cognitive performance, *Cognitive Technology* **14**(1): 14–25.
Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., and Wu, Y. (2023). How close is ChatGPT to human experts? comparison corpus, evaluation, and detection, *arXiv:2301.07597*.
Ha, V. and Musliner, D. J. (2002). Toward decision-theoretic CIRCA with application to real-time computer security control, *Proceedings of the AAAI Workshop on Real-Time Decision Support and Diagnosis Systems*, pp. 89–90.
Haarmann, H. J., Just, M. A., and Carpenter, P. A. (1997). Aphasic sentence comprehension as a resource deficit: A computational approach, *Brain and Language* **59**(1): 76–120.
Haikonen, P. O. (2007). *Robot Brains: Circuits and Systems for Conscious Machines*, John Wiley & Sons.
Halbrügge, M. (2007). Evaluating cognitive models and architectures, *Proceedings of the AAAI Workshop on Evaluating Architectures for Intelligence*, pp. 27–31.
Halbrügge, M. (2010). Keep it simple—A case study of model development in the context of the Dynamic Stocks and Flows (DSF) task, *Journal of Artificial General Intelligence* **2**(2): 38–51.
Hall, R. P. (1989). Computational approaches to analogical reasoning: A comparative analysis, *Artificial Intelligence* **39**(1): 39–120.
Halpern, M. (2006). The trouble with the Turing test, *The New Atlantis* (11): 42–63.
Hanna, R. (2023). How and why ChatGPT failed the Turing test, `https://againstprofphil.org/2023/01/15/how-and-why-chatgpt-failed-the-turing-test/`.
Hansen, E., Huntsberger, T., and Elkins, L. (2006). Autonomous maritime navigation: Developing autonomy skill sets for usvs, *Unmanned Systems Technology VIII*, Vol. 6230, International Society for Optics and Photonics, p. 62300U.
Hardwicke, T. E. et al. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal *Cognition*, *Royal Society Open Science* **5**(8): 180448.
Haring, K. S., Ragni, M., and Konieczny, L. (2012). A cognitive model of drivers attention, *Proceedings of the International Conference on Cognitive Modeling*, pp. 275–280.
Harman, G. (1976). Practical reasoning, *The Review of Metaphysics* **29**(3): 431–463.
Harnad, S. (1990). The symbol grounding problem, *Physica D: Nonlinear Phenomena* **42**(1–3): 335–346.
Harnad, S. (1991). Other bodies, other minds: A machine incarnation of an old philosophical problem, *Minds and Machines* **1**(1): 43–54.
Harrison, A. M. and Trafton, J. G. (2010). Cognition for action: An architectural account for "grounded interaction", *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 200–205.
Harrison, A. M., Schunn, C. D., et al. (2003). ACT-R/S: Look Ma, no "cognitive-map"!, *Proceedings of the International Conference on Cognitive Modeling*, pp. 129–134.
Hart, D. and Goertzel, B. (2008). Opencog: A software framework for integrative artificial general

intelligence, *Proceedings of the Artificial General Intelligence Conference*, pp. 468–472.
Hart, S. G., Dahn, D., Atencio, A., and Dalal, K. M. (2001). Evaluation and application of MIDAS v2.0, *Technical report*, SAE 2001-01-2648.
Hartley, R. and Pipitone, F. (1991). Experiments with the Subsumption architecture, *Proceedings of the International Conference on Robotics and Automation*, IEEE, pp. 1652–1658.
Hawkey, D. J., Amitay, S., and Moore, D. R. (2004). Early and rapid perceptual learning, *Nature Neuroscience* **7**(10): 1055–1056.
Hayes, P. J. (1981). The logic of frames, *Readings in Artificial Intelligence*, Elsevier, pp. 451–458.
Hayes-Roth, B. (1985a). A blackboard architecture for control, *Artificial Intelligence* **26**(3): 251–321.
Hayes-Roth, B. (1990). Architectural foundations for real-time performance in intelligent agents, *Journal of Real-Time Systems* **2**(1): 99–125.
Hayes-Roth, B. (1995). An architecture for adaptive intelligent systems, *Artificial Intelligence* **72**(1–2): 329–365.
Hayes-Roth, B. (1996). A domain-specific software architecture for a class of intelligent patient monitoring agents, *Journal of Experimental & Theoretical Artificial Intelligence* **8**(2): 149–171.
Hayes-Roth, B. and Collinot, A. (1994). A satisficing cycle for real-time reasoning in intelligent agents, *Expert Systems with Applications* **7**(1): 31–42.
Hayes-Roth, B., Washington, R., Hewett, R., Hewett, M., and Seiver, A. (1989). Intelligent monitoring and control, *Proceedings of the Joint Conference on Artificial Intelligence*, pp. 243–249.
Hayes-Roth, B., Washington, R., Ash, D., Hewett, R., Collinot, A., Vina, A., and Seiver, A. (1992). Guardian: A prototype intelligent agent for intensive-care monitoring, *Artificial Intelligence in Medicine* **4**(2): 165–185.
Hayes-Roth, B., Lalanda, P., Morignot, P., Pfleger, K., and Balabanovic, M. (1993). Plans and behavior in intelligent agents, *Technical Report 93*, Stanford University.
Hayes-Roth, B., Brownston, L., and Sincoff, E. (1995). Directed improvisation by computer characters, *Technical Report KSL-95-04*, Stanford University.
Hayes-Roth, F. (1985b). Rule-based systems, *Communications of the ACM* **28**(9): 921–932.
Hazy, T. E., Frank, M. J., and O'Reilly, R. C. (2006). Banishing the homunculus: Making working memory work, *Neuroscience* **139**(1): 105–118.
He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, *Proceedings of the International Conference on Computer Vision*, pp. 1026–1034.
Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., and Jennions, M. D. (2015). The extent and consequences of p-hacking in science, *PLoS Biology* **13**(3): e1002106.
Hebb, D. O. (1949). *Organization of Behavior*, Wiley, New York.
Hélie, S. and Sun, R. (2010). Incubation, insight, and creative problem solving: A unified theory and a connectionist model, *Psychological Review* **117**(3): 994–1024.
Hélie, S. and Sun, R. (2014). An integrative account of memory and reasoning phenomena, *New Ideas in Psychology* **35**: 36–52.
Hellendoorn, V. J. and Sawant, A. A. (2021). The growing cost of deep learning for source code, *Communications of the ACM* **65**(1): 31–33.
Hellendoorn, V. J., Sutton, C., Singh, R., Maniatis, P., and Bieber, D. (2019). Global relational models of source code, *Proceedings of the International Conference on Learning Representations*.
Hendler, J. (2008). Avoiding another AI winter, *IEEE Intelligent Systems* **23**(2): 2–4.
Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. (2021). Natural adversarial examples, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271.
Henninger, A. E., Jones, R. M., and Chown, E. (2002). Behaviors that emerge from emotion and cognition: A first evaluation, *Proceedings of the International Joint Conference on Autonomous Agents*, pp. 321–328.
Herd, S. A., Banich, M. T., and O'Reilly, R. C. (2006). Neural mechanisms of cognitive control: An integrative model of Stroop task performance and fMRI data, *Journal of Cognitive Neuroscience* **18**(1): 22–32.
Herd, S., Szabados, A., Vinokurov, Y., Lebiere, C., Cline, A., and O'Reilly, R. C. (2014). Integrating theories of motor sequencing in the SAL hybrid architecture, *Biologically Inspired Cognitive Architectures* **8**: 100–108.
Hernandez-Orallo, J. (2000). Beyond the Turing test, *Journal of Logic, Language and Information* **9**(4): 447–466.
Hernández-Orallo, J. (2017). Evaluation in artificial intelligence: From task-oriented to ability-oriented measurement, *Artificial Intelligence Review* **48**(3): 397–447.

Hernández-Orallo, J. and Dowe, D. L. (2010). Measuring universal intelligence: Towards an anytime intelligence test, *Artificial Intelligence* **174**(18): 1508–1539.

Hertz, J., Krogh, A., and Palmer, R. G. (2018). *Introduction to the Theory of Neural Computation*, CRC Press.

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. (2017). Deep learning scaling is predictable, empirically, *arXiv:1712.00409*.

Hexmoor, H., Lammens, J., and Shapiro, S. C. (1992). An autonomous agent architecture for integrating perception and acting with grounded, embodied symbolic reasoning, *Technical Report 92-21*, State University of New York at Buffalo.

Hicks, R. D. (1907). *Aristotle De Anima*, Cambridge University Press.

Hilario, M. (1997). An overview of strategies for neurosymbolic integration, *in* R. Sun and F. Alexandre (eds), *Connectionist-Symbolic Integration: From Unified to Hybrid Approaches*, Lawrence Erlbaum Associates, pp. 13–36.

Hilgard, E. R. and Marquis, D. G. (1940). *Conditioning and Learning*, Appleton-Century-Crofts.

Hingston, P. (2009). A Turing test for computer game bots, *IEEE Transactions on Computational Intelligence and AI in Games* **1**(3): 169–186.

Hingston, P. (2010). A new design for a Turing test for bots, *Proceedings of the IEEE Conference on Computational Intelligence and Games*, pp. 345–350.

Hinton, G. (2022). The forward-forward algorithm: Some preliminary investigations, *arXiv:2212.13345*.

Hinton, G. E. and McClelland, J. (1987). Learning representations by recirculation, *Neural Information Processing Systems*, pp. 358–366.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Computation* **9**(8): 1735–1780.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve, *The Journal of Physiology* **117**(4): 500–544.

Hofstader, D. (1979). *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books, Inc.

Hofstadter, D. (1994). How could a Copycat ever be creative?, *in* T. Dartnall (ed.), *Artificial Intelligence and Creativity*, Springer, pp. 405–424.

Hofstadter, D. R. (1982). The Turing test: A coffee-house conversation, *in* D. Hofstadter and D. Dennett (eds), *The Mind's I: Fantasies and Reflections on Self and Soul*, Penguin Books, pp. 69–95.

Hofstadter, D. R. (1984). The Copycat project: An experiment in nondeterminism and creative analogies, *Technical Report 755*, MIT.

Hofstadter, D. R. (2001). Analogy as the core of cognition, *in* D. Gentner, K. J. Holyoak, and B. N. Kokinov (eds), *The analogical mind: Perspectives from cognitive science*, MIT Press, pp. 499–538.

Hofstadter, D. R. and Mitchell, M. (1994). The Copycat project: A model of mental fluidity and analogy-making, *in* K. J. Holyoak and J. A. Barnde (eds), *Analogical Connections*, Ablex Publishing, pp. 31–112.

Holland, O., Diamond, A., Marques, H. G., Mitra, B., and Devereux, D. (2013). Real and apparent biological inspiration in cognitive architectures, *Biologically Inspired Cognitive Architectures* **3**: 105–116.

Holleman, G. A., Hooge, I. T., Kemner, C., and Hessels, R. S. (2020). The "real-world approach" and its problems: A critique of the term ecological validity, *Frontiers in Psychology* **11**: 721.

Holyoak, K. J. and Spellman, B. A. (1993). Thinking, *Annual Review of Psychology* **44**(1): 265–315.

Hopfield, J. J. and Tank, D. W. (1986). Computing with neural circuits: A model, *Science* **233**(4764): 625–633.

Horn, J. L. (1991). Measurement of intellectual capabilities: A review of theory, *in* K. S. McGrew, J. K. Werder, and R. W. Woodcock (eds), *Woodcock-Johnson technical manual*, pp. 197–232.

Horswill, I. and Brooks, R. A. (1988). Situated vision in a dynamic world: Chasing objects, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 796–800.

Houston, J. P. (1981). *Fundamentals of Learning and Memory*, 2nd edn, Academic Press.

Hoyle, M. A. and Lueg, C. (1997). Open Sesame!: A look at personal assistants, *Proceedings of the International Conference on the Practical Application of Intelligent Agents*, Vol. 51, p. 1997.

Hsu, S.-C. and Chien, C.-F. (2007). Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing, *International Journal of Production Economics* **107**(1): 88–103.

Huang, H.-M., Messina, E., et al. (1996). NIST-RCS and object-oriented methodologies of software engineering: A conceptual comparison, *Proceedings of the Conference on Intelligent Systems: A Semiotic Perspective*, IEEE.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex,

The Journal of Physiology **148**(3): 574–591.

Hudlicka, E. (2002). This time with feeling: Integrated model of trait and state effects on cognition and behavior, *Applied Artificial Intelligence* **16**(7-8): 611–641.

Hudlicka, E. (2005). A computational model of emotion and personality: Applications to psychotherapy research and practice, *Proceedings of the Annual CyberTherapy Conference: A Decade of Virtual Reality*.

Hudlicka, E. (2007). Reasons for emotions: Modeling emotions in integrated cognitive systems, *in* W. D. Gray (ed.), *Integrated Models of Cognition Systems*, Oxford University Press, pp. 263–278.

Hudlicka, E. (2010). Modeling cultural and personality biases in decision-making, *Proceedings of the International Conference on Applied Human Factors and Ergonomics*.

Hudlicka, E. and Broekens, J. (2009). Foundations for modelling emotions in game characters: Modelling emotion effects on cognition, *Proceedings of the International Conference on Affective Computing and Intelligent Interaction and Workshops*, pp. 1–6.

Hudlicka, E., Zacharias, G., and Psotka, J. (2000). Increasing realism of human agents by modeling individual differences: Methodology, architecture, and testbed, *Proceedings of the AAAI Fall Symposium*, pp. 53–59.

Hull, J. J. (1994). A database for handwritten text recognition research, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**(5): 550–554.

Hummel, J. E. and Holyoak, K. J. (1997). Distributed representations of structure: A theory of analogical access and mapping, *Psychological Review* **104**(3): 427–466.

Hummel, J. E. and Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization, *Psychological Review* **110**(2): 220–264.

Hummel, J. E. and Holyoak, K. J. (2005). Relational reasoning in a neurally plausible cognitive architecture: An overview of the LISA project, *Current Directions in Psychological Science* **14**(3): 153–157.

Hunt, E. and Luce, R. D. (1992). Soar as a world view, not a theory, *Behavioral and Brain Sciences* **15**(3): 447–448.

Huntsberger, T. (2011a). Cognitive architecture for mixed human-machine team interactions for space exploration, *Proceedings of the IEEE Conference on Aerospace*, IEEE, pp. 1–11.

Huntsberger, T. (2011b). Process algebra approach for action recognition in the maritime domain, *Proceedings of the International Conference on Intelligent Robots and Systems, Workshop on Advances in Marine Autonomy Research*.

Huntsberger, T. and Stoica, A. (2010). Envisioning cognitive robots for future space exploration, *in* J. J. Braun (ed.), *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Vol. 7710, p. 77100D.

Huntsberger, T., Aghazarian, H., Howard, A., and Trotz, D. C. (2011). Stereo vision–based navigation for autonomous surface vessels, *Journal of Field Robotics* **28**(1): 3–18.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. (2020). Compositionality decomposed: How do neural networks generalise?, *Journal of Artificial Intelligence Research* **67**: 757–795.

Hutchens, J. L. (1997). How to pass the Turing test by cheating, *Technical Report TR97-05*, University of Western Australia.

Ikle, M. and Goertzel, B. (2011). Nonlinear-dynamical attention allocation via information geometry, *International Conference on Artificial General Intelligence*, Springer, pp. 62–71.

Ingham, J. (1997). What is an agent?, *Technical Report 6/99*, University of Durham.

Ingrand, F. and Coutance, V. (1993). Procedural reasoning versus blackboard architecture for real-time reasoning, *Proceedings of the International Conference on Artificial Intelligence*.

Ingrand, F. F. and Georgeff, M. P. (1990). Managing deliberation and reasoning in real-time AI systems, *Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*, pp. 284–291.

Insa-Cabrera, J., Hernández-Orallo, J., Dowe, D. L., Espana, S., and Hernández-Lloreda, M. V. (2012). The anYnt project intelligence test: Lambda-one, *Proceedings of the AISB/IACAP Symposium "Revisiting Turing and His Test"*, pp. 20–27.

Ioannidis, J. P. (2012). Why science is not necessarily self-correcting, *Perspectives on Psychological Science* **7**(6): 645–654.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(11): 1254–1259.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons?, *IEEE Transactions on Neural Networks* **15**(5): 1063–1070.

Jakubovitz, D., Giryes, R., and Rodrigues, M. R. (2019). Generalization error in deep learning, *in* G. Kutyniok, R. Mathar, and P. Petersen (eds), *Compressed Sensing and Its Applications*, Springer, pp. 153–193.

Jiang, N., Chen, C., He, J., Meng, J., Pan, L., Su, S., and Zhu, X. (2023). Bio-robotics research for non-invasive myoelectric neural interfaces for upper-limb prosthetic control: A 10-year perspective review, *National Science Review* **10**(5): nwad048.

Jilk, D. J., Lebiere, C., O'Reilly, R. C., and Anderson, J. R. (2008). SAL: An explicitly pluralistic cognitive architecture, *Journal of Experimental and Theoretical Artificial Intelligence* **20**(3): 197–218.

John, B. E. (1993). A quantitative model of expert transcription typing, *Technical Report CMU-CS-93-120*, Carnegie Mellon University.

Johnson-Laird, P. N. (2010). Mental models and human reasoning, *Proceedings of the National Academy of Sciences* **107**(43): 18243–18250.

Johnson-Laird, P. N. and Byrne, R. M. J. (1993). Précis of *Deduction*, *Behavioral and Brain Sciences* **16**(2): 323–333.

Johnson-Laird, P. N. and Shafir, E. (1993). The interaction between reasoning and decision making: An introduction, *Cognition* **49**(1–2): 1–9.

Jonas, E. and Kording, K. P. (2017). Could a neuroscientist understand a microprocessor?, *PLoS Computational Biology* **13**(1): 1–24.

Jones, J. L., Seiger, B. A., and Flynn, A. M. (1998). *Mobile Robots: Inspiration to Implementation*, CRC Press.

Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., and Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation, *AI Magazine* **20**(1): 27–41.

Jonsdottir, G. R. and Thórisson, K. R. (2013). A distributed architecture for real-time dialogue and on-task learning of efficient co-operative turn-taking, *in* M. Rojc and N. Campbell (eds), *Coverbal Synchrony in Human-Machine Interaction*, CRC Press, pp. 293–323.

Joshi, H. and Ustun, V. (2023). Augmenting cognitive architectures with large language models, *Proceedings of the AAAI Fall Symposium Series*, pp. 281–285.

Joshi, S., Schermerhorn, P., Khardon, R., and Scheutz, M. (2012). Abstract planning for reactive robots, *Proceedings of the International Conference on Robotics and Automation*, pp. 4379–4384.

Just, M. A. and Carpenter, P. A. (1985). Cognitive coordinate systems: Accounts of mental rotation and individual differences in spatial ability, *Psychological Review* **92**(2): 137–172.

Just, M. A. and Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory, *Psychological Review* **99**(1): 122.

Just, M. A. and Varma, S. (2002). A hybrid architecture for working memory: Reply to MacDonald and Christiansen (2002), *Psychological Review* **109**(1): 55–65.

Just, M. A. and Varma, S. (2007). The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition, *Cognitive, Affective, & Behavioral Neuroscience* **7**(3): 153–191.

Kachergis, G., Wyatte, D., O'Reilly, R. C., de Kleijn, R., and Hommel, B. (2014). A continuous-time neural model for sequential action, *Philosophical Transactions of the Royal Society B: Biological Sciences* **369**(1655): 20130623.

Kakumanu, P., Makrogiannis, S., and Bourbakis, N. (2007). A survey of skin-color modeling and detection methods, *Pattern Recognition* **40**(3): 1106–1122.

Kandel, E. R., Koester, J. D., Mack, S. H., and Siegelbaum, S. A. (2021). *Principles of Neural Science*, 6th edn, McGraw-Hill.

Kandemir, C., Handley, H. A., and Thompson, D. (2018). A workload model to evaluate distracters and driver's aids, *International Journal of Industrial Ergonomics* **63**: 18–36.

Kane, J. S. and Paquin, M. J. (1993). POPART: Partial optical implementation of Adaptive Resonance Theory 2, *IEEE Transactions on Neural Networks* **4**(4): 695–702.

Kapoor, S. and Narayanan, A. (2022). Leakage and the reproducibility crisis in machine-learning-based science, *Patterns* **4**: 100804.

Kautz, H. (2022). The third AI summer: AAAI Robert S. Engelmore memorial lecture, *AI Magazine* **43**(1): 105–125.

Kawamura, K. (2023). A perspective on cognitive robot research and development, *International Journal of Humanoid Robotics* p. 2350023.

Kawamura, K., Bagchi, S., Iskarous, M., Pack, R. T., and Saad, A. (1993). An intelligent robotic aid system for human services, *Proceedings of the Conference on Intelligent Robots in Factory, Field, Space, and Service*, pp. 413–420.

Kawamura, K., Bagchi, S., Iskarous, M., and Bishay, M. (1995). Intelligent robotic systems in service of the disabled, *IEEE Transactions on Rehabilitation Engineering* **3**(1): 14–21.

Kawamura, K., Peters II, R. A., Bodenheimer, R. E., Sarkar, N., Park, J., Clifton, C. A., Spratley, A. W., and Hambuchen, K. A. (2004). A parallel distributed cognitive control system for a humanoid robot, *International Journal of Humanoid Robotics* **1**(1): 65–93.

Kawamura, K., Gordon, S. M., Ratanaswasd, P., Erdemir, E., and Hall, J. F. (2008). Implementation of cognitive control for a humanoid robot, *International Journal of Humanoid Robotics* **5**(4): 547–586.

Kedar, S. T. and McKusick, K. B. (1992). There is no free lunch: Tradeoffs in the utility of learned knowledge, *Technical Report MS 269-2*, NASA Ames Reseach Center.

Kelemen, A., Liang, Y., Kozma, R., and Franklin, S. (2003). Optimizing intelligent agent's constraint satisfaction with neural networks, *in* A. Abraham, L. C. Jain, and J. Kacprzyk (eds), *Recent Advances in Intelligent Paradigms and Applications*, Springer, pp. 255–272.

Kelkar, S. (2022). Between AI and learning science: The evolution and commercialization of intelligent tutoring systems, *IEEE Annals of the History of Computing* **44**(1): 20–30.

Kelley, H. J. (1960). Gradient theory of optimal flight paths, *Journal of the American Rocket Society* **30**(10): 947–954.

Kelley, T. D. (2006). Developing a psychologically inspired cognitive architecture for robotic control: The Symbolic and Subsymbolic Robotic Intelligence Control System (SS-RICS), *International Journal of Advanced Robotic Systems* **3**(3): 219–222.

Kelley, T., Avery, E., and McGhee, S. (2011). The perception problem and the impact on robotics and computer vision, *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Vol. 8064, International Society for Optics and Photonics, p. 80640B.

Kellman, P. J. and Massey, C. M. (2013). Perceptual learning, cognition, and expertise, *in* B. H. Ross (ed.), *Psychology of Learning and Motivation*, Elsevier, pp. 117–165.

Kennedy, M. B. (2016). Synaptic signaling in learning and memory, *Cold Spring Harbor Perspectives in Biology* **8**(2): a016824.

Kennedy, W. G. and Patterson, R. E. (2012). Modeling intuitive decision making in ACT-R, *Proceedings of the International Conference on Cognitive Modeling*, pp. 1–6.

Kennedy, W. G. and Trafton, J. G. (2007). Long-term symbolic learning, *Cognitive Systems Research* **8**(3): 237–247.

Kennedy, W. G., Bugajska, M. D., Adams, W., Schultz, A. C., and Trafton, J. G. (2008). Incorporating mental simulation for a more effective robotic teammate, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1300–1305.

Kennedy, W. G., Bugajska, M. D., Harrison, A. M., and Trafton, J. G. (2009). "Like-me" simulation as an effective and cognitively plausible basis for social robotics, *International Journal of Social Robotics* **1**(2): 181–194.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art, *Information Fusion* **14**(1): 28–44.

Kieras, D. E. (2005). Fidelity issues in cognitive architectures for HCI modelling: Be careful what you wish for, *Proceedings of the International Conference on Human Computer Interaction*, pp. 22–27.

Kieras, D. E. (2007). Control of cognition, *in* W. D. Gray (ed.), *Integrated Models of Cognitive Systems*, Oxford University Press, pp. 327–355.

Kieras, D. E. and Hornof, A. J. (2014). Towards accurate and practical predictive models of active-vision-based visual search, *Proceedings of Conference on Human Factors in Computing Systems*, pp. 3875–3884.

Kieras, D. E. and Meyer, D. E. (1994). The EPIC architecture for modeling human information-processing and performance: A brief introduction, *Technical Report TR-94/ONR-EPIC-1*, University of Michigan.

Kieras, D. E. and Meyer, D. E. (1996). EPIC architecture principles of operation, *Technical report*, University of Michigan. `https://web.eecs.umich.edu/~kieras/docs/EPICtutorial2004/EPICPrinOp.pdf`.

Kieras, D. E., Wood, S. D., and Meyer, D. E. (1997). Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task, *ACM Transactions on Computer-Human Interaction* **4**(3): 230–275.

Kieras, D. E., Meyer, D. E., Mueller, S., and Seymour, T. (1998). Insights into working memory from the perspective of the EPIC architecture for modeling skilled perceptual-motor and cognitive human performance, *Technical Report TR-98/ONR-EPIC-10*, University of Michigan.

Kieras, D. E., Hornof, A., and Zhang, Y. (2015). Visual search of displays of many objects: Modeling detailed eye movement effects with improved EPIC, *Proceedings of the International Conference on Cognitive Modeling*, Vol. 55–60.

Kieras, D. E., Wakefield, G. H., Thompson, E. R., Iyer, N., and Simpson, B. D. (2016). Modeling two-channel speech processing with the EPIC cognitive architecture, *Topics in Cognitive Science* **8**(1): 291–304.

Kietzmann, T. C., McClure, P., and Kriegeskorte, N. (2018). Deep neural networks in computational neuroscience, *BioRxiv:10.1101/133504* p. 133504.

Kim, J. W., Koubek, R. J., and Ritter, F. E. (2007). Investigation of procedural skills degradation from different modalities, *Proceedings of the International Conference on Cognitive Modeling*, pp. 255–260.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. (2019). The (un)reliability of saliency methods, *in* W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller (eds), *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280.

Kirk, J. R. and Laird, J. E. (2016). Learning general and efficient representations of novel games through interactive instruction, *Proceedings of the Annual Conference on Advances in Cognitive Systems*.

Kiryazov, K., Petkov, G., Grinberg, M., Kokinov, B., and Balkenius, C. (2006). The interplay of analogy-making with active vision and motor control in anticipatory robots, *in* M. V. Butz, O. Sigaud, G. Pezzulo, and G. Baldassarre (eds), *Anticipatory Behavior in Adaptive Learning Systems*, Springer, pp. 233–253.

Kiyonaga, A. and Egner, T. (2013). Working memory as internal attention: Toward an integrative account of internal and external selection processes, *Psychonomic Bulletin & Review* **20**: 228–242.

Klenk, M. and Forbus, K. D. (2007). Measuring the level of transfer learning by an AP Physics problem-solver, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 446–451.

Klenk, M., Forbus, K. D., Tomai, E., Kim, H., and Kyckelhahn, B. (2005). Solving everyday physical reasoning problems by analogy using sketches, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 209–215.

Knowles, K., Witbrock, M., Dobbie, G., and Yogarajan, V. (2023). A proposal for a language model based cognitive architecture, *Proceedings of the AAAI Fall Symposium Series*, pp. 295–301.

Kočiskỳ, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. (2018). The NarrativeQA reading comprehension challenge, *Transactions of the Association for Computational Linguistics* **6**: 317–328.

Koenig, N. and Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 2149–2154.

Kokinov, B. (1988). Associative memory-based reasoning: How to represent and retrieve cases, *Proceedings of the International Conference on Artificial Intelligence: Methodology, Systems, Applications*, pp. 51–57.

Kokinov, B. (1990). Associative memory-based reasoning: Some experimental results, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 741–749.

Kokinov, B. (1994a). Flexibility versus efficiency: The DUAL answer, *in* P. Jorrand and V. Sgurev (eds), *Artificial intelligence: Methodology, Systems, Applications*, pp. 321–330.

Kokinov, B. (1994b). A hybrid model of reasoning by analogy, *in* K. Holyoak and J. Barnden (eds), *Advances in Connectionist and Neural Computation Theory*, Vol. 2, Ablex, pp. 247–318.

Kokinov, B. (2013). Micro-level hybridization in the cognitive architecture DUAL, *in* R. Sun and F. Alexandre (eds), *Connectionist-Symbolic Integration*, Psychology Press, pp. 197–208.

Kokinov, B. and French, R. M. (2003). Computational models of analogy-making, *Encyclopedia of Cognitive Science* **1**: 113–118.

Kokinov, B., Hristova, P., and Petkov, G. (2004). Does irrelevant information play a role in judgment?, *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 26, pp. 720–725.

Kokinov, B., Petkov, G., and Petrova, N. (2007). Context-sensitivity of human memory: Episode connectivity and its influence on memory reconstruction, *Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context*, pp. 317–329.

Kokinov, B., Grinberg, M., Petkov, G., and Kiryazov, K. (2008). Anticipation by analogy, *in* G. Pezzulo, M. V. Butz, C. Castelfranchi, and R. Falcone (eds), *The Challenge of Anticipation*, Springer, pp. 185–213.

Könik, T. and Laird, J. E. (2006). Learning goal hierarchies from structured observations and expert annotations, *Machine Learning* **64**(1–3): 263–287.

Konolige, K., Myers, K., Ruspini, E., and Saffiotti, A. (1997). The Saphira architecture: A design for autonomy, *Journal of Experimental & Theoretical Artificial Intelligence* **9**(2–3): 215–235.

Konolige, K. G., Gutmann, S., Guzzoni, D., Ficklin, R. W., and Nicewarner, K. E. (1999). A mobile robot sense net, *Sensor Fusion and Decentralized Control in Robotic Systems II*, Vol. 3839, International Society for Optics and Photonics, pp. 74–83.

Kortenkamp, D., MacMahon, M., Ryan, D., Bonasso, R. P., and Moreland, L. (1998). Applying a layered control architecture to a free-flying space camera, *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, pp. 188–194.

Kotseruba, I. (2016). *Visual attention in dynamic environments and its application to playing online games*, Master's thesis, York University.

Kotseruba, I. and Tsotsos, J. K. (2017). Star-rt: Visual attention for real-time video game playing, *arXiv:1711.09464*.

Kotseruba, I. and Tsotsos, J. K. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications, *Artificial Intelligence Review* **53**: 17–94.

Kotseruba, I., Papagelis, M., and Tsotsos, J. K. (2021). Industry and academic research in computer vision, *arXiv:2107.04902*.

Krawczyk, D. C., Holyoak, K. J., and Hummel, J. E. (2004). Structural constraints and object similarity in analogical mapping and inference, *Thinking & Reasoning* **10**(1): 85–104.

Krichmar, J. L. (2000). Experience-dependent perceptual categorization in a behaving real-world device, *Proceedings of the International Conference on the Simulation of Adaptive Behavior*, pp. 41–50.

Krichmar, J. L. (2012). Design principles for biologically inspired cognitive robotics, *Biologically Inspired Cognitive Architectures* **1**: 73–81.

Krichmar, J. L. and Edelman, G. M. (2002). Machine psychology: Autonomous behavior, perceptual categorization and conditioning in a brain-based device, *Cerebral Cortex* **12**(8): 818–830.

Krichmar, J. L. and Edelman, G. M. (2005). Brain-based devices for the study of nervous systems and the development of intelligent machines, *Artificial Life* **11**(1-2): 63–77.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems* **25**: 1097–1105.

Krotkov, E. and Simmons, R. (1996). Perception, planning, and control for autonomous walking with the ambler planetary rover, *The International Journal of Robotics Research* **15**(2): 155–180.

Krueger, J. I., Heck, P. R., Evans, A. M., and DiDonato, T. E. (2020). Social game theory: Preferences, perceptions, and choices, *European Review of Social Psychology* **31**(1): 222–253.

Kruijne, W. and Tsotsos, J. K. (2011). Visuo-cognitive routines: Reinterpreting the theory of visual routines as a framework for visual cognition, *Technical Report TR CSE-2011-05*, York University.

Kubose, T. T., Holyoak, K. J., and Hummel, J. E. (2002). The role of textual coherence in incremental analogical mapping, *Journal of Memory and Language* **47**(3): 407–435.

Kuznetsov, V., Mohri, M., and Syed, U. (2014). Multi-class deep boosting, *Advances in Neural Information Processing Systems* **27**: 2501–2509.

Kyllonen, P. C. and Shute, V. J. (1988). Taxonomy of learning skills—Interim technical report for the period February 1986-1987, *Technical Report AFHRL-TP-87-39*, Air Force Human Resources Laboratory.

Lachman, S. J. (1997). Learning is a process: Toward an improved definition of learning, *The Journal of Psychology* **131**(5): 477–480.

Laird, J. E. (1991). Preface for special section on integrated cognitive architectures, *ACM SIGART Bulletin* **2**(4): 12–13.

Laird, J. E. (2001). It knows what you're going to do: Adding anticipation to a Quakebot, *Proceedings of the International Conference on Autonomous Agents*, pp. 385–392.

Laird, J. E. (2008). Extending the Soar cognitive architecture, *Frontiers in Artificial Intelligence and Applications* **171**: 224.

Laird, J. E. (2012a). *The Soar Cognitive Architecture*, MIT Press.

Laird, J. E. (2012b). The Soar cognitive architecture, *AISB Quarterly* (134): 1–4.

Laird, J. E. (2022a). An analysis and comparison of ACT-R and Soar, *Proceedings of the Annual Conference on Advances in Cognitive Systems*.

Laird, J. E. (2022b). Introduction to Soar, *arXiv:2205.03854*.

Laird, J. E. and Jones, R. M. (1999). Building advanced autonomous AI systems for large scale real time simulations, *Technical report*, University of Michigan.

Laird, J. E. and Mohan, S. (2014). A case study of knowledge integration across multiple memories in Soar, *Biologically Inspired Cognitive Architectures* **8**: 93–99.

Laird, J. E. and Rosenbloom, P. S. (2014). The evolution of the Soar cognitive architecture, *in* D. M. Steier and T. T. Mitchell (eds), *Mind Matters*, Psychology Press, pp. 1–50.

Laird, J. E. and Wray III, R. E. (2010). Cognitive architecture requirements for achieving AGI, *Proceedings of the Conference on Artificial General Intelligence*, pp. 79–84.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence, *Artificial Intelligence* **33**(1): 1–64.

Laird, J. E., Yager, E. S., Hucka, M., and Tuck, C. M. (1991). Robo-Soar: An integration of external interaction, planning, and learning using Soar, *Robotics and Autonomous Systems* **8**(1-2): 113–129.

Laird, J. E., Coulter, K. J., Jones, R. M., Kenny, P. G., Koss, F., and Nielsen, P. E. (1998). Integrating intelligent computer generated forces in distributed simulations: TacAir-Soar in STOW–97, *Proceedings of the Simulation Interoperability Workshop*.

Laird, J. E., Derbinsky, N., and Voigt, J. (2011). Performance evaluation of declarative memory systems in Soar, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*, pp. 33–40.

Laird, J. E., Kinkade, K. R., Mohan, S., and Xu, J. Z. (2012). Cognitive robotics using the Soar cognitive architecture, *Proceedings of the AAAI Conference on Artificial Intelligence Workshops*, pp. 46–54.

Laird, J. E., Lebiere, C., and Rosenbloom, P. S. (2017). A standard model of the mind: Toward a common computational framework across artificial intelligence, cognitive science, neuroscience, and robotics, *AI Magazine* **38**(4): 13–26.

Lake, B. and Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, *Proceedings of the International Conference on Machine Learning*, pp. 2873–2882.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people, *Behavioral and Brain Sciences* **40**: e253.

Landauer, T. K. (1986). How much do people remember? Some estimates of the quantity of learned information in long-term memory, *Cognitive Science* **10**(4): 477–493.

Landgrebe, J. and Smith, B. (2022). *Why Machines Will Never Rule the World: Artificial Intelligence Without Fear*, Routledge.

Lane, P. C. and Gobet, F. (2012). A theory-driven testing methodology for developing scientific software, *Journal of Experimental & Theoretical Artificial Intelligence* **24**(4): 421–456.

Lane, P. C., Gobet, F., and Smith, R. L. (2008). Attention mechanisms in the CHREST cognitive architecture, *International Workshop on Attention in Cognitive Systems*, pp. 183–196.

Lane, P. C., Sykes, A., and Gobet, F. (2019). Combining low-level perception with expectations in CHREST, *Proceedings of European Cognitive Science Conference*, pp. 205–210.

Langley, P. (1996a). An abstract computational model of learning selective sensing skills, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 385–390.

Langley, P. (1996b). *Elements of Machine Learning*, Morgan Kaufmann.

Langley, P. (2006a). Cognitive architectures and general intelligent systems, *AI Magazine* **27**(2): 33–44.

Langley, P. (2006b). Intelligent behavior in humans and machines, *Proceedings of the Dartmouth Artificial Intelligence Conference: The Next Fifty Years (AI50)*.

Langley, P., Choi, D., and Rogers, S. (2005a). Interleaving learning, problem-solving, and execution in the ICARUS architecture, *Technical report*, Stanford University.

Langley, P., Magnani, L., Schunn, C., and Thagard, P. (2005b). An extended theory of human problem solving, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1242–1247.

Langley, P., Choi, D., and Rogers, S. (2009). Acquisition of hierarchical reactive skills in a unified cognitive architecture, *Cognitive Systems Research* **10**(4): 316–332.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization, *Journal de Physiologie et de Pathologie Générale* **9**: 620–635. English translation in (Brunel et al., 2007).

Larkin, M., Eatough, V., and Osborn, M. (2011). Interpretative phenomenological analysis and embodied, active, situated cognition, *Theory & Psychology* **21**(3): 318–337.

Lassila, O. (1990). Frames or objects, or both? Workshop notes from the Eighth National Conference on Artificial Intelligence: Object-Oriented Programming in AI, *Technical Report HTKK-TKO-B67*, Helsinki University of Technology.

Lathrop, S. D. and Laird, J. E. (2007). Towards incorporating visual imagery into a cognitive architecture, *Proceedings of the International Conference on Cognitive Modeling*, pp. 1–6.

Lazarus, R. (1994). Universal antecedents of the emotions, *in* P. Ekman and R. Davison (eds), *The Nature of Emotions: Fundamental Questions*, Oxford University Press, pp. 163–171.

Leavitt, M. L. and Morcos, A. (2020). Towards falsifiable interpretability research, *Advances in Neural Information Processing Systems Workshop on ML Retrospectives, Surveys, & Meta-Analyses*.

Lebiere, C. (2006). Constrained functionality: Application of the ACT-R cognitive architecture to the AMBR modeling comparison, *Modeling Human Behavior with Integrated Cognitive Architectures*, Psychology Press, pp. 81–130.

Lebiere, C., Biefeld, E., Archer, R., Archer, S., Allender, L., and Kelley, T. D. (2002). IMPRINT/ACT-R: Integration of a task network modeling architecture with a cognitive architecture and its application to human error modeling, *in* M. J. Chinni (ed.), *Military,*

Government and Aerospace Simulation, Vol. 34, pp. 13–19.

Lebiere, C., Pirolli, P., Thomson, R., Paik, J., Rutledge-Taylor, M., Staszewski, J., and Anderson, J. R. (2013). A functional model of sensemaking in a neurocognitive architecture, *Computational Intelligence and Neuroscience* **2013**: 921695.

LeCun, Y. and Cortes, C. (1998). The MNIST database of handwritten digits, `http://yann.lecun.com/exdb/mnist/`.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Handwritten digit recognition with a back-propagation network, *Advances in Neural Information Processing Systems*, pp. 396–404.

Lee, M. D., Criss, A. H., Devezer, B., Donkin, C., Etz, A., Leite, F. P., Matzke, D., Rouder, J. N., Trueblood, J. S., White, C. N., et al. (2019). Robust modeling in cognitive science, *Computational Brain & Behavior* **2**: 141–153.

Legg, S. and Hutter, M. (2007a). A collection of definitions of intelligence, *in* B. Goertzel and P. Wang (eds), *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms*, IOS press, pp. 17–24.

Legg, S. and Hutter, M. (2007b). Universal intelligence: A definition of machine intelligence, *Minds and Machines* **17**: 391–444.

Legrenzi, P., Girotto, V., and Johnson-Laird, P. N. (1993). Focussing in reasoning and decision making, *Cognition* **49**(1–2): 37–66.

Lemoine, B. (2022). Is LaMDA Sentient?—An Interview, https://cajundiscordian.medium.com/is-lamda-sentient-an-interview-ea64d916d917.

Lerner, J. S., Li, Y., Valdesolo, P., and Kassam, K. S. (2015). Emotion and decision making, *Annual Review of Psychology* **66**(1): 799–823.

Lesser, E., Schaeps, T., Haikonen, P. O., and Jorgensen, C. (2008). Associative neural networks for machine consciousness: Improving existing AI technologies, *Proceedings of the IEEE Convention of Electrical and Electronics Engineers*, pp. 11–15.

Levesque, H. J. (1986). Knowledge representation and reasoning, *Annual Review of Computer Science* **1**(1): 255–287.

Lewis, A. and Smith, D. (1993). Defining higher order thinking, *Theory Into Practice* **32**(3): 131–137.

Leydesdorff, L. and Goldstone, R. L. (2014). Interdisciplinarity at the journal and specialty level: The changing knowledge bases of the journal cognitive science, *Journal of the Association for Information Science and Technology* **65**(1): 164–177.

Licato, J., Sun, R., and Bringsjord, S. (2014). Using a hybrid cognitive architecture to model children's errors in an analogy task, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 857–862.

Lighthill, J. (1973). Artificial intelligence: A general survey, *Artificial Intelligence: A Paper Symposium*, Science Research Council, pp. 1–73.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning, *Nature Communications* **7**(1): 13276.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020). Backpropagation and the brain, *Nature Reviews Neuroscience* **21**(6): 335–346.

Lishner, D. A. (2015). A concise set of core recommendations to improve the dependability of psychological research, *Review of General Psychology* **19**(1): 52–68.

Lisman, J. E. and Jensen, O. (2013). The theta-gamma neural code, *Neuron* **77**(6): 1002–1016.

Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., and Murphy, K. (2018). Progressive neural architecture search, *Proceedings of the European Conference on Computer Vision*, pp. 19–34.

Ljungberg, M. and Lucas, A. (1992). The OASIS Air Traffic Management System, *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pp. 185–190.

Lloyd-Kelly, M., Lane, P. C., and Gobet, F. (2014). The effects of bounding rationality on the performance and learning of CHREST agents in Tileworld, *Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 149–162.

Lloyd-Kelly, M., Gobet, F., and Lane, P. C. (2015). Piece of mind: Long-term memory structure in ACT-R and CHREST, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1422–1428.

Lloyd-Kelly, M., Gobet, F., and Lane, P. C. (2016). Under pressure: How time-limited cognition explains statistical learning by 8-month old infants, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1475–1480.

Loeser, J. D. and Melzack, R. (1999). Pain: An overview, *The Lancet* **353**(9164): 1607–1609.

Long, L. N. (2017). A model for temperament and emotions on robots, *Proceedings of the International Conference on Applied Human Factors and Ergonomics*, pp. 3–13.

Long, L. N., Kelley, T. D., and Avery, E. S. (2015). An emotion and temperament model for cognitive

mobile robots, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*, pp. 66–73.

Lorenz, K. (1935). Der Kumpan in der Umwelt des Vogels. Der Artgenosse als auslösendes Moment sozialer Verhaltungsweisen [The companion in the bird's world. The fellow-member of the species as releasing factor of social behavior], *Journal für Ornithologie* **35**: 137–213.

Loui, R. P. (1993). Analogy, decision, and theory-formation as defeasible reasoning, *Technical Report WUCS-93-39*, Washington University.

Love, B. C. (2021). Levels of biological plausibility, *Philosophical Transactions of the Royal Society B* **376**(1815): 20190632.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**: 91–110.

Lu, Z.-L., Williamson, S. J., and Kaufman, L. (1992). Behavioral lifetime of human auditory sensory memory predicted by physiological measures, *Science* **258**(5088): 1668–1670.

Luck, M. and d'Inverno, M. (2003). Unifying agent systems, *Annals of Mathematics and Artificial Intelligence* **37**(1): 131–167.

Luger, G. F. (2009). *Artificial intelligence: Structures and Strategies for Complex Problem Solving*, 6th edn, Addison-Wesley.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models, *Neural Networks* **10**(9): 1659–1671.

Mackworth, A. K. (1993). On seeing robots, *in* A. Basu (ed.), *Computer Vision: Systems, Theory and Applications*, World Scientific, pp. 1–13.

Macpherson, F. (2011). Taxonomising the senses, *Philosophical Studies* **153**(1): 123–142.

Madl, T. and Franklin, S. (2012). A LIDA-based model of the attentional blink, *Proceedings of the International Conference on Computational Modeling*.

Madl, T. and Franklin, S. (2015). Constrained incrementalist moral decision making for a biologically inspired cognitive architecture, *in* R. Trappl (ed.), *A Construction Manual for Robots' Ethical Systems*, Springer, pp. 137–153.

Madl, T., Franklin, S., Chen, K., Montaldi, D., and Trappl, R. (2016). Towards real-world capable spatial memory in the LIDA cognitive architecture, *Biologically Inspired Cognitive Architectures* **16**: 87–104.

Maes, P. (1989). How to do the right thing, *Connection Science* **1**(3): 291–323.

Maes, P. (1991). The agent network architecture (ANA), *ACM SIGART Bulletin* **2**(4): 115–120.

Maffei, G., Santos-Pata, D., Marcos, E., Sánchez-Fibla, M., and Verschure, P. F. (2015). An embodied biologically constrained model of foraging: From classical and operant conditioning to adaptive real-world behavior in DAC-X, *Neural Networks* **72**: 88–108.

Malaviya, M., Sucholutsky, I., Oktar, K., and Griffiths, T. L. (2022). Can humans do less-than-one-shot learning?, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 997–1003.

Malone, J. C. (1975). William James and B. F. Skinner: Behaviorism, reinforcement, and interest, *Behaviorism* **3**(2): 140–151.

Malone, N., Faust, A., Rohrer, B., Lumia, R., Wood, J., and Tapia, L. (2014). Efficient motion-based task learning for a serial link manipulator, *Transactions on Control and Mechanical Systems* **3**(1): 25–35.

Manso, L., Bachiller, P., Bustos, P., Núnez, P., Cintas, R., and Calderita, L. (2010). RoboComp: A tool-based robotics framework, *Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 251–262.

Manso, L. J., Calderita, L., Bustos, P., García Guzmán, J., Martinez Muñoz, M., Fernández Rebollo, F., Romero Garcés, A., and Bandera, A. (2014). A general-purpose architecture to control mobile robots, *Proceedings of the Workshop of Physical Agents*, pp. 105–116.

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Toward an integration of deep learning and neuroscience, *Frontiers in Computational Neuroscience* **10**: 94.

Marcus, G. (2018). Deep learning: A critical appraisal, *arXiv:1801.00631*.

Marcus, G. (2022). Deep learning is hitting a wall, *Nautilus*. `https://nautil.us/deep-learning-is-hitting-a-wall-238440/`.

Marewski, J. N. and Mehlhorn, K. (2011). Using the ACT-R architecture to specify 39 quantitative process models of decision making, *Judgment and Decision Making* **6**: 439–519.

Marfil, R., Romero-Garcés, A., Bandera, J. P., Manso, L. J., Calderita, L. V., Bustos, P., Bandera, A., Garcia-Polo, J., Fernandez, F., and Voilmy, D. (2019). Perceptions or actions? Grounding how agents interact within a software architecture for cognitive robotics, *Cognitive Computation* **12**: 1–19.

Marinier III, R. P., Laird, J. E., and Lewis, R. L. (2009). A computational unification of cognitive behavior and emotion, *Cognitive Systems Research* **10**(1): 48–69.

Markovitch, S. and Scott, P. D. (1988). The role of forgetting in learning, *Proceedings of the International Conference on Machine Learning*, pp. 459–465.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W. H. Freeman and Company.

Marshall, J. B. (2002). Metacat: A self-watching cognitive architecture for analogy-making, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 631–636.

Marshall, J. B. (2006). A self-watching model of analogy-making and perception, *Journal of Experimental and Theoretical Artificial Intelligence* **18**(3): 267–307.

Martens, S., Carpenter, G. A., and Gaudiano, P. (1998a). Neural sensor fusion for spatial visualization on a mobile robot, *Sensor Fusion and Decentralized Control in Robotic Systems*, Vol. 3523, International Society for Optics and Photonics, pp. 100–111.

Martens, S., Gaudiano, P., and Carpenter, G. A. (1998b). Mobile robot sensor integration with fuzzy ARTMAP, *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 307–312.

Martínez-Fernández, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., Vollmer, A. M., and Wagner, S. (2022). Software engineering for AI-based systems: A survey, *ACM Transactions on Software Engineering and Methodology* **31**(2): 1–59.

Martínez-Plumed, F., Avin, S., Brundage, M., Dafoe, A., hÉigeartaigh, S. Ó., and Hernández-Orallo, J. (2018). Between progress and potential impact of AI: The neglected dimensions, *arXiv:1806.00610*.

Martínez-Plumed, F., Barredo, P., hÉigeartaigh, S. Ó., and Hernández-Orallo, J. (2021). Research community dynamics behind popular AI benchmarks, *Nature Machine Intelligence* **3**(7): 581–589.

Mather, G. (2016). *Foundations of Sensation and Perception*, Routledge.

Mathews, Z., Lechón, M., Calvo, J. B., Dhir, A., Duff, A., Bermúdez i Baida, S., and Verschure, P. F. (2009). Insect-like mapless navigation based on head direction cells and contextual learning using chemo-visual sensors, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 2243–2250.

Mathews, Z., Sergi Bermúdez i Badia, S., and Verschure, P. F. (2012). PASAR: An integrated model of prediction, anticipation, sensation, attention and response for artificial sensorimotor systems, *Information Sciences* **186**(1): 1–19.

Mayo, D., Lu, D., Zhang, C., Cummings, J., Lin, X., Katz, B., Glass, J. R., and Barbu, A. (2022). Growing ObjectNet: Adding speech, VQA, occlusion, and measuring dataset difficulty, *Proceedings of the International Conference on Machine Learning Workshop Shift Happens*.

McCarthy, J. (1959). Programs with common sense, *Proceedings of the Teddington Conference on the Mechanization of Thought Processsses*, pp. 75–91.

McCarthy, J. (1978). History of LISP, *in* R. L. Wexelblat (ed.), *History of Programming Languages*, Academic Press, pp. 173–185.

McCarthy, J., Minsky, M. L., Rochester, N., and Shannon, C. E. (2006). A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955, *AI Magazine* **27**(4): 12–14.

McCauley, L. and Franklin, S. (2002). A large-scale multi-agent system for Navy personnel distribution, *Connection Science* **14**(4): 371–385.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics* **5**: 115–133.

McDermott, D. (1992). Robot planning, *AI Magazine* **13**(2): 55–79.

McDermott, D., Waldrop, M. M., Chandrasekaran, B., McDermott, J., and Schank, R. (1985). The dark ages of AI: A panel discussion at AAAI-84, *AI Magazine* **6**(3): 122–122.

McGrew, K. S. (2005). The Cattell-Horn-Carroll theory of cognitive abilities: Past, present, and future, *in* D. P. Flanagan and P. L. Harrison (eds), *Contemporary Intellectual Assessment: Theories, Tests, and Issues*, The Guilford Press, pp. 136–181.

McGrew, K. S. (2023). Carroll's three-stratum (3S) cognitive ability theory at 30 years: Impact, 3S-CHC theory clarification, structural replication, and cognitive–achievement psychometric network analysis extension, *Journal of Intelligence* **11**(2): 32.

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices, *Nature* **264**(5588): 746–748.

McHugh, C. and Way, J. (2018). What is reasoning?, *Mind* **127**(505): 167–196.

McKinstry, J. L., Seth, A. K., Edelman, G. M., and Krichmar, J. L. (2008). Embodied models of delayed neural responses: Spatiotemporal categorization and predictive motor control in brain based devices, *Neural networks* **21**(4): 553–561.

Medeiros, A. A. (1998). A survey of control architectures for autonomous mobile robots, *Journal of the Brazilian Computer Society* **4**(3): 35–43.

Medsker, L. R. and Bailey, D. L. (1992). Models and guidelines for integrating expert systems and neural networks, *in* A. Kandel and G. Langholz (eds), *Hybrid Architectures for Intelligent*

Systems, CRC Press, pp. 153–171.
Mehrani, P. and Tsotsos, J. K. (2021). Early recurrence enables figure border ownership, *Vision Research* **186**: 23–33.
Mehrani, P. and Tsotsos, J. K. (2023). Learning a model of shape selectivity in V4 cells reveals shape encoding mechanisms in the brain, *Journal of Neuroscience* **43**(22): 4129–4143.
Melton, A. W. (ed.) (1964). *Categories of Human Learning*, Academic Press.
Menager, D. and Choi, D. (2016). A robust implementation of episodic memory for a cognitive architecture, *Proceedings of the Annual Meeting of the Cognitive Science Society*.
Mendoza, C., Bachiller, P., Bandera, A., and Bustos, P. (2018). Visual attention mechanisms revisited, *Proceedings of the Workshop of Physical Agents*, Springer, pp. 100–114.
Merel, J., Botvinick, M., and Wayne, G. (2019). Hierarchical motor control in mammals and machines, *Nature Communications* **10**(1): 1–12.
Merkle, R. C. (1989). Energy limits to the computational power of the human brain, `http://www.ralphmerkle.com/brainLimits.html`.
Meyer, D. E. and Kieras, D. E. (1994). EPIC computational models of psychological refractory-period effects in human multiple-task performance, *Technical Report TR-94/ONR-EPIC-2*, University of Michigan.
Meyer, D. E. and Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part I. Basic mechanisms, *Psychological Review* **104**(1): 3–65.
Michalski, R. S. (1986). Understanding the nature of learning: Issues and research directions, *Technical Report 938*, University of Illinois.
Miksch, S., Cheng, K., and Hayes-Roth, B. (1997). An intelligent assistant for patient health care, *Proceedings of the International Conference on Autonomous Agents*, pp. 458–465.
Miller, D. P. and Slack, M. G. (1991). Global symbolic maps from local navigation, *Proceedings of the National Conference on Artificial Intelligence*, pp. 750–755.
Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information, *Psychological Review* **63**(2): 81.
Miller, G. A. (2003). The cognitive revolution: A historical perspective, *Trends in Cognitive Sciences* **7**(3): 141–144.
Miller, R. B. (1967). Task taxonomy: Science or technology?, *Ergonomics* **10**(2): 167–176.
Mininger, A. and Laird, J. E. (2016). Interactively learning strategies for handling references to unseen or unknown objects, *Proceedings of the Annual Conference on Advances in Cognitive Systems*.
Minsky, M. (1974). A framework for representing knowledge, *Technical Report 306*, MIT.
Minsky, M. (1987). The Society of Mind, *The Personalist Forum*, Vol. 3, University of Illinois Press, pp. 19–32.
Minsky, M. (1988). *Society of Mind*, Simon and Schuster.
Minsky, M. and Papert, A. S. (1969). *Perceptrons*, MIT Press.
Minton, S. (1990). Quantitative results concerning the utility of explanation-based learning, *Artificial Intelligence* **42**(2-3): 363–391.
Mitchell, D. K. (2000). Mental workload and ARL workload modeling tools, *Technical Report ARL-TN-161*, Army Research Laboratory.
Mitchell, D. K. (2009). Workload analysis of the crew of the Abrams V2 SEP: Phase I baseline IMPRINT model, *Technical Report ADA508882*, Army Research Lab.
Mitchell, M. (2023). How do we know how smart AI systems are?, *Science* **381**(6654): eadj5957.
Mitchell, M. and Hofstadter, D. R. (1990). The emergence of understanding in a computer model of concepts and analogy-making, *Physica D: Nonlinear Phenomena* **42**(1-3): 322–334.
Mitchell, T. M. (2006). The discipline of machine learning, *Technical Report CMU-ML-06-108*.
Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view, *Machine Learning* **1**(1): 47–80.
Mnih, V. et al. (2015). Human-level control through deep reinforcement learning, *Nature* **518**(7540): 529–533.
Mohan, S., Mininger, A. H., Kirk, J. R., and Laird, J. E. (2012). Acquiring grounded representations of words with situated interactive instruction, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, pp. 113–130.
Momennejad, I. (2022). A rubric for human-like agents and NeuroAI, *Philosophical Transactions of the Royal Society B* **378**(1869): 20210446.
Moor, J. (2006). The Dartmouth College Artificial Intelligence Conference: The next fifty years, *AI Magazine* **27**(4): 87–87.
Moore, J. (2011). Behaviorism, *The Psychological Record* **61**: 449–463.
Moors, A. (2014). Flavors of appraisal theories of emotion, *Emotion Review* **6**(4): 303–307.
Moravec, H. (1998). When will computer hardware match the human brain, *Journal of Evolution*

and Technology **1**: 1–10.

Moreno, R. A. and de Miguel, A. S. (2006). A machine consciousness approach to autonomous mobile robotics, *Proceedings of the AAAI International Cognitive Robotics Workshop*, pp. 111–118.

Moreno, R. A., Espino, A. L., and de Miguel, A. S. (2007). Modeling consciousness for autonomous robot exploration, *in* J. Mira and J. R. Álvarez (eds), *Bio-Inspired Modeling of Cognitive Tasks*, Springer, pp. 51–60.

Morey, R. D. and Lakens, D. (2017). Why most of psychology is statistically unfalsifiable, `https://zenodo.org/record/838685`.

Morrison, R. G., Holyoak, K. J., and Troung, B. (2001). Working-memory modularity in analogical reasoning, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Morrison, R. G., Krawczyk, D. C., Holyoak, K. J., Hummel, J. E., Chow, T. W., Miller, B. L., and Knowlton, B. J. (2004). A neurocomputational model of analogical reasoning and its breakdown in frontotemporal lobar degeneration, *Journal of Cognitive Neuroscience* **16**(2): 260–271.

Morrison, R. G., Doumas, L. A., and Richland, L. E. (2006). The development of analogical reasoning in children: A computational account, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Morse, A. F., De Greeff, J., Belpeame, T., and Cangelosi, A. (2010). Epigenetic robotics architecture (ERA), *IEEE Transactions on Autonomous Mental Development* **2**(4): 325–339.

Moto-Oka, T. and Stone, H. S. (1984). Fifth-generation computer systems: A Japanese project, *Computer* **17**: 6–13.

Mueller, R. J. and Mueller, C. L. (1995). The cognitive revolution and the computer. Version 2, *Technical Report ED 386 169*.

Mueller, S. T. (2010). A partial implementation of the BICA Cognitive Decathlon using the Psychology Experiment Building Language (PEBL), *International Journal of Machine Consciousness* **2**(2): 273–288.

Mueller, S. T., Jones, M., Minnery, B. S., and Hiland, J. M. (2007). The BICA Cognitive Decathlon: A test suite for biologically-inspired cognitive agents, *Proceedings of the Conference on Behavior Representation in Modeling and Simulation*, pp. 418–450.

Murphy, K. N., Juberts, M., Legowik, S. A., Nashman, M., Schneiderman, H., Scott, H. A., and Szabo, S. (1994). Ground vehicle control at NIST: From teleoperation to autonomy, *Proceedings of the Seventh Annual Workshop on Space Operations Applications and Research*, pp. 137–142.

Musliner, D. J., Durfee, E. H., and Shin, K. G. (1993). CIRCA: A cooperative intelligent real-time control architecture, *IEEE Transactions on Systems, Man, and Cybernetics* **23**(6): 1561–1574.

Musliner, D. J., Shin, K. G., and Durfee, E. H. (1994). Automating the design of real-time reactive systems, *Proceedings of Symposium on AI in Real-Time Control*.

Myers Jr, J. P. and Yamakoshi, K. (2020). The Japanese Fifth Generation Computing Project: A brief overview, *Journal of Computing Sciences in Colleges* **36**(2): 53–60.

Mylopoulos, J. (1980). An overview of knowledge representation, *ACM SIGART Bulletin* (74): 5–12.

Nakauchi, Y. and Simmons, R. (1999). Social behavioral robot that stands in line, *Proceedings of the International Conference on Systems, Man, and Cybernetics*, IEEE, pp. 993–998.

Nasir, I., Iqbal, M., and Raza, S. A. (2022). XML-based Descriptive Language for Cognitive Architectures, *International Journal of Computational and Innovative Sciences* **1**(2): 33–46.

Nathan, A., Grimberg, J., and Rhodes, A. (2024). Gen AI: Too much spend, too little benefit?, `https://www.goldmansachs.com/intelligence/pages/gs-research/gen-ai-too-much-spend-too-little-benefit/report.pdf`. [Accessed July 19, 2024].

National Research Council (1999). *Funding a Revolution: Government Support for Computing Research*, National Academies Press.

Navarro, R. (2009). The optical design of the human eye: A critical review, *Journal of Optometry* **2**: 3–18.

Nestor, A. and Kokinov, B. (2004). Towards active vision in the dual cognitive architecture, *Information Theories and Applications* **11**: 9–15.

Newell, A. (1962). Some problems of basic organization in problem-solving programs, *Technical Report RM-3283-PR*, The RAND Corporation.

Newell, A. (1973a). Production systems: Models of control structures, *Proceedings of the Annual Carnegie Symposium on Cognition*, Elsevier, pp. 463–526.

Newell, A. (1973b). You can't play 20 questions with nature and win, *in* W. G. Chase (ed.), *Visual Information Processing. Proceedings of the Eighth Annual Carnegie Symposium on Cognition*, Elsevier, pp. 283–308.

Newell, A. (1980). Physical symbol systems, *Cognitive Science* **4**(2): 135–183.

Newell, A. (1981). Review of Nils Nillson, Principles of Artificial Intelligence, *Contemporary Psychology* **26**: 50–51.

Newell, A. (1982). The knowledge level, *Artificial Intelligence* **18**(1): 87–127.

Newell, A. (1990). *Unified Theories of Cognition*, Harvard University Press.

Newell, A. and Simon, H. A. (1961). Computer simulation of human thinking, *Science* **134**(3495): 2011–2017.

Newell, A. and Simon, H. A. (1972). *Human Problem Solving*, Prentice-Hall.

Newell, A. and Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search, *Communications of the ACM* **19**: 113–126.

Newell, A., Shaw, J. C., and Simon, H. A. (1958). Elements of a theory of human problem solving, *Psychological Review* **65**(3): 151–166.

Newell, A., Shaw, J. C., and Simon, H. A. (1959). Report on a general problem solving program, *Technical Report P-1584*, Carnegie Institute of Technlogy.

Newell, A., Rosenbloom, P. S., and Laird, J. E. (1989). Symbolic architectures for cognition, *in* M. Posner (ed.), *Foundations of Cognitive Science*, MIT Press, pp. 93–131.

Newman, S. D., Carpenter, P. A., Varma, S., and Just, M. A. (2003). Frontal and parietal participation in problem solving in the Tower of London: fMRI and computational modeling of planning and high-level perception, *Neuropsychologia* **41**(12): 1668–1682.

Nield, T. (2019). Is deep learning already hitting its limitations? And is another AI winter coming?, `https://towardsdatascience.com/is-deep-learning-already-hitting-its-limitations-c81826082ac3`.

Nii, P. H. (1986). Blackboard systems, *Technical Report STAN-CS-16-123*, Stanford University.

Nilsson, N. J. (2005). Human-level artificial intelligence? Be serious!, *AI Magazine* **26**(4): 68–75.

Norman, D. A. and Shallice, T. (1986). Attention to action, *in* R. K. Davidson, G. E. Schwartz, and D. Shapiro (eds), *Consciousness and self-regulation*, Springer, pp. 1–18.

Norman, E., Pfuhl, G., Sæle, R. G., Svartdal, F., Låg, T., and Dahl, T. I. (2019). Metacognition in psychology, *Review of General Psychology* **23**(4): 403–424.

Nosek, B. A. et al. (2022). Replicability, robustness, and reproducibility in psychological science, *Annual Review of Psychology* **73**: 719–748.

Núñez, R., Allen, M., Gao, R., Rigoli, C. M., Relaford-Doyle, J., and Semenuks, A. (2019). What happened to cognitive science?, *Nature Human Behaviour* **3**(8): 782–791.

Núñez, R., Allen, M., Gao, R., Miller Rigoli, C., Relaford-Doyle, J., and Semenuks, A. (2020). For the sciences they are a-changin': A response to commentaries on Núñez et al.'s (2019) "What happened to cognitive science?", *Topics in Cognitive Science* **12**(3): 790–803.

Nuxoll, A., Laird, J. E., and James, M. (2004). Comprehensive working memory activation in Soar, *International Conference on Cognitive Modeling*, pp. 226–230.

Nuxoll, A. M. and Laird, J. E. (2007). Extending cognitive architecture with episodic memory, *Proceedings of the National Conference on Artificial Intelligence*, pp. 1560–1564.

Nyamsuren, E. and Taatgen, N. (2013a). The synergy of top-down and bottom-up attention in complex task: Going beyond saliency models, *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 35, pp. 3181–3186.

Nyamsuren, E. and Taatgen, N. A. (2013b). Pre-attentive and attentive vision module, *Cognitive Systems Research* **24**: 62–71.

Oberauer, K. (2019). Working memory and attention—A conceptual analysis and review, *Journal of Cognition* **2**(1): 1–23.

Oden, G. C. (1987). Concept, knowledge, and thought, *Annual Review of Psychology* **38**(1): 203–227.

Ogasawara, G. H. and Russell, S. J. (1993). Planning using multiple execution architectures, *Proceedings of the Joint Conference on Artificial Intelligence*, pp. 1394–1401.

Olah, C., Mordvintsev, A., and Schubert, L. (2017). Feature visualization, *Distill*. https://distill.pub/2017/feature-visualization.

Olson, M. H. and Hergenhahn, B. R. (2016). *Introduction to Theories of Learning*, 9th edn, Psychology Press.

Omnivision (2022). OVB0B 200 megapixel product brief, `https://www.ovt.com/products/ovb0b/`.

O'Neill, K., Bridewell, W., and Bello, P. (2018). Time-based resource sharing in ARCADIA, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 828–833.

Open Science Collaboration (2015). Estimating the reproducibility of psychological science, *Science* **349**(6251): aac4716.

O'Reilly, R. C. (1996). *The Leabra Model of Neural Interactions and Learning in the Neocortex*, PhD thesis, Carnegie Mellon University.

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition, *Trends in Cognitive Sciences* **2**(11): 455–462.

O'Reilly, R. C. and Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*, MIT Press.

O'Reilly, R. C., Frank, M. J., Hazy, T. E., and Watz, B. (2007). PVLV: The primary value and learned value Pavlovian learning algorithm, *Behavioral Neuroscience* **121**(1): 31–49.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., and Contributors (2012). *Computational Cognitive Neuroscience*, Online Book, 4th Edition, URL: `https://CompCogNeuro.org`.

O'Reilly, R. C., Wyatte, D., Herd, S., Mingus, B., and Jilk, D. J. (2013). Recurrent processing during object recognition, *Frontiers in Psychology* **4**: 124.

Ortiz Jr, C. L. (2016). Why we need a physically embodied Turing test and what it might look like, *AI Magazine* **37**(1): 55–62.

Ortony, A. and Turner, T. J. (1990). What's basic about basic emotions?, *Psychological Review* **97**(3): 315–331.

Oteiza, P. and Baldwin, M. W. (2021). Evolution of sensory systems, *Current Opinion in Neurobiology* **71**: 52–59.

Öztürk, P. (2005). Modules, layers, hierarchies, and loops where artificial intelligence meets ethology and neuroscience—In context of action selection, *Proceedings of the International Conference on Mechanisms, Symbols, and Models Underlying Cognition*, pp. 16–26.

Öztürk, P. (2009). Levels and types of action selection: The action selection soup, *Adaptive Behavior* **17**(6): 537–554.

O'Reilly, R. C., Hazy, T. E., and Herd, S. A. (2016). The Leabra cognitive architecture: How to play 20 principles with nature, *in* S. E. F. Chipman (ed.), *Oxford Handbook of Cognitive Science*, Vol. 91, Oxford University Press, pp. 91–116.

Pack, R. T., Wilkes, M., Biswas, G., and Kawamura, K. (1997). Intelligent machine architecture for object-based system integration, *Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 151–157.

Page, M. (2000). Connectionist modelling in psychology: A localist manifesto, *Behavioral and Brain Sciences* **23**(4): 443–512.

Paisner, M., Cox, M., Maynord, M., and Perlis, D. (2014). Goal-driven autonomy for cognitive systems, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 2085–2090.

Paivio, A. (1975). Imagery and synchronic thinking, *Canadian Psychological Review/Psychologie Canadienne* **16**(3): 147.

Paluszek, M. and Thomas, S. (2020). *Practical MATLAB Deep Learning*, Springer.

Papert, S. (1988). One AI or many?, *Daedalus* pp. 1–14.

Paraense, A. L., Raizer, K., de Paula, S. M., Rohmer, E., and Gudwin, R. R. (2016). The cognitive systems toolkit and the CST reference cognitive architecture, *Biologically Inspired Cognitive Architectures* **17**: 32–48.

Paritosh, P. and Marcus, G. (2016). Toward a comprehension challenge, using crowdsourcing as a tool, *AI Magazine* **37**(1): 23–30.

Parker, D. B. (1982). Learning logic, *Technical Report S81-64*, Stanford University.

Paszke, A. et al. (2019). PyTorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems* pp. 8024–8035.

Pauli, W. M. and O'Reilly, R. C. (2008). Attentional control of associative learning—A possible role of the central cholinergic system, *Brain Research* **1202**: 43–53.

Peebles, D. (2019). Modelling mental imagery in the ACT-R cognitive architecture, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 2550–2556.

Peebles, D. and Jones, C. (2014). A model of object location memory, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 2747–2752.

Peng, J., Peters, A., Ao, X., and Srikaew, A. (2003). Grasping a waving object for a humanoid robot using a biologically-inspired active vision system, *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pp. 115–120.

Peters, R. A., Hambuchen, K. E., Kawamura, K., and Wilkes, D. M. (2001a). The sensory ego-sphere as a short-term memory for humanoids, *Proceedings of the IEEE International Conference on Humanoid Robots*, pp. 451–459.

Peters, R. A., Kawamura, K., Wilkes, D. M., Hambuchen, K. A., Rogers, T. E., and Alford, W. A. (2001b). ISAC humanoid: An architecture for learning and emotion, *Proceedings of the International Conference on Humanoid Robots*.

Petkov, G., Vankov, I., and Kokinov, B. (2011). Unifying deduction, induction, and analogy by the AMBR model, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 574–579.

Pew, R. W. and Mavor, A. S. (eds) (1998). *Modeling Human and Organizational Behavior—Application to Military Simulations*, National Academy Press.

Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*, MIT press.

Pfeifer, R. and Scheier, C. (1997). Sensory-motor coordination: The metaphor and beyond, *Robotics and Autonomous Systems* **20**(2-4): 157–178.

Pfeiffer, M. and Pfeil, T. (2018). Deep learning with spiking neurons: Opportunities and challenges, *Frontiers in Neuroscience* **12**: 774.

Pfeiffer, U. J., Timmermans, B., Bente, G., Vogeley, K., and Schilbach, L. (2011). A non-verbal Turing test: Differentiating mind from machine in gaze-based social interaction, *PloS One* **6**(11): e27591.

Pfleger, K. and Hayes-Roth, B. (1997). An introduction to blackboard-style systems organization, *Technical Report KSL-98-03*, Stanford University.

Piaget, J. (1952). *The Origin of Intelligence in the Child*, Routledge.

Piccinini, G. (2000). Turing's rules for the imitation game, *Minds and Machines* **10**(4): 573–582.

Pinar Saygin, A., Cicekli, I., and Akman, V. (2000). Turing test: 50 years later, *Minds and Machines* **10**(4): 463–518.

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Larochelle, H. (2021). Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program), *The Journal of Machine Learning Research* **22**(1): 7459–7478.

Pinker, S. (1984). Visual cognition: An introduction, *Cognition* **18**(1-3): 1–63.

Pirjanian, P. (1999). Behavior coordination mechanisms—State-of-the-art, *Technical Report IRIS-99-375*, University of Southern California.

Place, U. T. (1992). Eliminative connectionism: Its implications for a return to an empiricist/behaviorist linguistics, *Behavior and Philosophy* pp. 21–35.

Pokahr, A., Braubach, L., and Lamersdorf, W. (2005). Jadex: A BDI reasoning engine, *in* R. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni (eds), *Multi-agent programming: Languages, Platforms and Applications*, Springer, pp. 149–174.

Pollack, J. B. (1989). Connectionism: Past, present, and future, *Artificial Intelligence Review* **3**(1): 3–20.

Pollock, J. (1999). Natural deduction, `https://www.johnpollock.us/ftp/OSCAR-web-page/PAPERS/Natural-Deduction.pdf`.

Pollock, J. L. (1987). Defeasible reasoning, *Cognitive Science* **11**(4): 481–518.

Pollock, J. L. (1989). OSCAR: A general theory of rationality, *Journal of Experimental & Theoretical Artificial Intelligence* **1**(3): 209–226.

Pollock, J. L. (1993). Planning in OSCAR, *Proceedings of the AAAI Spring Symposium*, pp. 163–164.

Pollock, J. L. (2000). Defeasible reasoning in OSCAR, *Proceedings of the International Workshop on Non-Monotonic Reasoning*.

Pollock, J. L. and Hosea, D. (1995). OSCAR-MDA: An artificially intelligent advisor for emergency room medicine, *Technical report*, University of Arizona.

Posner, M. I., Nissen, M. J., and Klein, R. M. (1976). Visual dominance: An information-processing account of its origins and significance, *Psychological Review* **83**(2): 157–171.

Post, E. L. (1943). Formal reductions of the general combinatorial decision problem, *American Journal of Mathematics* **65**(2): 197–215.

Prescott, T. J. (2008). Action selection, *Scholarpedia* **3**(2): 2705.

Priti, S. and Miyake, A. (1997). Models of working memory: An introduction, *in* A. Miyake and P. Shah (eds), *Models of Working Memory: Mechanisms of Active Maintenance and Executive Control*, Cambridge University Press, pp. 1–27.

Purchase, H. C., Archambault, D., Kobourov, S., Nöllenburg, M., Pupyrev, S., and Wu, H.-Y. (2020). The Turing test for graph drawing algorithms, *Proceedings of the International Symposium on Graph Drawing and Network Visualization*, Springer, pp. 466–481.

Pyke, A., West, R. L., and LeFevre, J.-A. (2007). How readers retrieve referents for nouns in real time: A memory-based model of context effects on referent accessibility, *Proceedings of the International Conference on Cognitive Modeling*, pp. 7–12.

Pynadath, D. V., Rosenbloom, P. S., Marsella, S. C., and Li, L. (2013). Modeling two-player games in the Sigma graphical cognitive architecture, *International Conference on Artificial General Intelligence*, Springer, pp. 98–108.

Pynadath, D. V., Rosenbloom, P. S., and Marsella, S. C. (2014). Reinforcement learning for adaptive theory of mind in the Sigma cognitive architecture, *Proceedings of the International Conference on Artificial General Intelligence*, Springer, pp. 143–154.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: An open-source Robot Operating System, *Proceedings of the International Conference on Robotics and Automation Workshop on Open Source Software*, Vol. 3, p. 5.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities, *Behavioral Science* **12**(5): 410–430.

Raff, E. (2019). A step toward quantifying independently reproducible machine learning research, *Advances in Neural Information Processing Systems* **32**: 5485–5495.

Rahmanzadehgervi, P., Bolton, L., Taesiri, M. R., and Nguyen, A. T. (2024). Vision language models are blind, *arXiv:2407.06581*.

Raji, I. D., Kumar, I. E., Horowitz, A., and Selbst, A. (2022). The fallacy of AI functionality, *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pp. 959–972.

Rakover, S. S. (2020). Why has the field of psychology not developed like the natural sciences?, *The Journal of Mind and Behavior* **41**(3/4): 247–266.

Ramamurthy, U., Baars, B. J., and Franklin, S. (2006). LIDA: A working model of cognition, *Proceedings of the International Conference on Cognitive Modeling*, pp. 244–249.

Rao, A. S. and Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture, *Technical Report 14*, Australian Artificial Intelligence Institute.

Rao, A. S. and Georgeff, M. P. (1995). BDI agents: From theory to practice, *Proceedings of the International Conference on Multiagent Systems*, pp. 312–319.

Ras, G., van Gerven, M., and Haselager, P. (2018). Explanation methods in deep learning: Users, values, concerns and challenges, *in* H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven (eds), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, Springer, pp. 19–36.

Rashid, M., Sulaiman, N., Abdul PP Majeed, A., Musa, R. M., Bari, B. S., and Khatun, S. (2020). Current status, challenges, and possible solutions of EEG-based brain-computer interface: A comprehensive review, *Frontiers in Neurorobotics* **14**: 25.

Raut, R. V., Snyder, A. Z., and Raichle, M. E. (2020). Hierarchical dynamics as a macroscopic organizing principle of the human brain, *Proceedings of the National Academy of Sciences* **117**(34): 20890–20897.

Redgrave, P., Prescott, T. J., and Gurney, K. (1999). The basal ganglia: A vertebrate solution to the selection problem?, *Neuroscience* **89**(4): 1009–1023.

Reeke, G. N., Sporns, O., and Edelman, G. M. (1990). Synthetic neural modeling: The "Darwin" series of recognition automata, *Proceedings of the IEEE* **78**(9): 1498–1530.

Reisenzein, R., Hudlicka, E., Dastani, M., Gratch, J., Hindriks, K., Lorini, E., and Meyer, J.-J. C. (2013). Computational modeling of emotion: Toward improving the inter-and intradisciplinary exchange, *IEEE Transactions on Affective Computing* **4**(3): 246–266.

Renoult, L., Irish, M., Moscovitch, M., and Rugg, M. D. (2019). From knowing to remembering: The semantic–episodic distinction, *Trends in Cognitive Sciences* **23**(12): 1041–1057.

Rhodes, M. G. (2019). Metacognition, *Teaching of Psychology* **46**(2): 168–175.

Ribeiro, D., Hinrichs, T., Crouse, M., Forbus, K., Chang, M., and Witbrock, M. (2019). Predicting state changes in procedural text using analogical question answering, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, pp. 1–6.

Richards, B. A. et al. (2019). A deep learning framework for neuroscience, *Nature Neuroscience* **22**(11): 1761–1770.

Rips, L. J. (1983). Cognitive processes in propositional reasoning, *Psychological Review* **90**(1): 38.

Rips, L. J. (1990). Reasoning, *Annual Review of Psychology* **41**(1): 321–353.

Ritter, F. E. (accepted 2022). *Design Patterns for Modeling and HCI*, Oxford University Press.

Ritter, F. E. and Serdiuk, S. (2024). Towards a comprehensive summary of senses for cognitive architectures, *Proceedings of the International Conference on Cognitive Modeling*.

Ritter, F. E., Avraamides, M., and Councill, I. G. (2002). Validating changes to a cognitive architecture to more accurately model the effects of two example behavior moderators, *Proceedings of the Computer-Generated Forces and Behavior Representation Conference*, pp. 29–40.

Ritter, F. E., Shadbolt, N. R., Elliman, D., Young, R. M., Gobet, F., and Baxter, G. D. (2003). Techniques for modeling human performance in synthetic environments: A supplementary review, *Technical report*, Human Systems Information Analysis Center.

Ritter, F. E., Bittner, J. L., Kase, S. E., Evertsz, R., Pedrotti, M., and Busetta, P. (2012). CoJACK: A high-level cognitive architecture with demonstrations of moderators, variability, and implications for situation awareness, *Biologically Inspired Cognitive Architectures* **1**: 2–13.

Ritter, F. E., Tehranchi, F., and Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition, *Wiley Interdisciplinary Reviews: Cognitive Science* **10**(3): e1488.

Ritter, S., Anderson, J. R., Koedinger, K. R., and Corbett, A. (2007). Cognitive Tutor: Applied research in mathematics education, *Psychonomic Bulletin & Review* **14**: 249–255.

Roberts, M. J. (1993). Human reasoning: Deduction rules or mental models, or both?, *The Quarterly Journal of Experimental Psychology Section A* **46**(4): 569–589.

Roberts, S. and Pashler, H. (2000). How persuasive is a good fit? A comment on theory testing, *Psychological Review* **107**(2): 358–367.

Roelfsema, P. R. and de Lange, F. P. (2016). Early visual cortex as a multiscale cognitive blackboard, *Annual Review of Vision Science* **2**: 131–151.

Rogers, T. and Wilkes, M. (2000). The Human Agent: A work in progress toward human-humanoid interaction, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 864–869.

Rohrer, B. (2007a). S-Learning: A biomimetic algorithm for learning, memory, and control in robots, *Proceedings of the IEEE International Conference on Neural Engineering*, pp. 148–151.

Rohrer, B. (2011). Biologically inspired feature creation for multi-sensory perception, *Proceedings of the Annual Meeting of the BICA Society*, pp. 305–313.

Rohrer, B. (2012). BECCA: Reintegrating AI for natural world interaction, *Proceedings of AAAI Spring Symposium*.

Rohrer, B. (2013). BECCA version 0.4.5 User's Guide.

Rohrer, B., Bernard, M., Morrow, J. D., Rothganger, F., and Xavier, P. (2009). Model-free learning and control in a mobile robot, *Proceedings of the IEEE International Conference on Natural Computation*, pp. 566–572.

Rohrer, B. R. (2007b). Robust performance of autonomous robots in unstructured environments, *Technical Report SAND2007-7568C 521473*, Sandia National Laboratory.

Romero-Garcés, A., Calderita, L. V., Martínez-Gómez, J., Bandera, J. P., Marfil, R., Manso, L. J., Bandera, A., and Bustos, P. (2015a). Testing a fully autonomous robotic salesman in real scenarios, *Proceedings of the International Conference on Autonomous Robot Systems and Competitions*, IEEE, pp. 124–130.

Romero-Garcés, A., Calderita, L. V., Martınez-Gómez, J., Bandera, J. P., Marfil, R., Manso, L. J., Bustos, P., and Bandera, A. (2015b). The cognitive architecture of a robotic salesman, *Proceedings of the Conference of Spanish Association for Artificial Intelligence*, pp. 16–24.

Romero, O. J., Zimmerman, J., Steinfeld, A., and Tomasic, A. (2023). Synergistic integration of large language models and cognitive architectures for robust AI: An exploratory analysis, *Proceedings of the AAAI Fall Symposium Series*, Vol. 2, pp. 396–405.

Rose, A. (1973). *Vision: Human and Electronic*, Plenum Press.

Rosen, B. R. and Savoy, R. L. (2012). fMRI at 20: Has it changed the world?, *NeuroImage* **62**(2): 1316–1324.

Rosenberg-Lee, M., Lovett, M. C., and Anderson, J. R. (2009). Neural correlates of arithmetic calculation strategies, *Cognitive, Affective, & Behavioral Neuroscience* **9**(3): 270–285.

Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review* **65**(6): 386–408.

Rosenbloom, P. S. and Aasman, J. (1990). Knowledge level and inductive uses of chunking (ebl), *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 821–827.

Rosenbloom, P. S., Demski, A., and Ustun, V. (2015a). Efficient message computation in Sigma's graphical architecture, *Biologically Inspired Cognitive Architectures* **11**: 1–9.

Rosenbloom, P. S., Gratch, J., and Ustun, V. (2015b). Towards emotion in Sigma: From appraisal to attention, *International Conference on Artificial General Intelligence*, pp. 142–151.

Rosenbloom, P. S., Demski, A., and Ustun, V. (2016). The Sigma cognitive architecture and system: Towards functionally elegant grand unification, *Journal of Artificial General Intelligence* **7**(1): 1–103.

Rosenfeld, A., Biparva, M., and Tsotsos, J. K. (2018). Priming neural networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2011–2020.

Rosenfeld, J. S. (2021). *Scaling Laws for Deep Learning*, PhD thesis, MIT.

Rosenschein, S. J. and Kaelbling, L. P. (1989). Integrating planning and reactive control, *Proceedings of the NASA Conference on Space Telerobotics*, Vol. 2, pp. 359–366.

Rossi, J. S. (1990). Statistical power of psychological research: What have we gained in 20 years?, *Journal of Consulting and Clinical Psychology* **58**(5): 646–656.

Rougier, N. P. and O'Reilly, R. C. (2002). Learning representations in a gated prefrontal cortex model of dynamic task switching, *Cognitive Science* **26**(4): 503–520.

Rousseau, D. and Hayes-Roth, B. (1997). Improvisational synthetic actors with flexible personalities, *Technical Report KSL 97-10*, Stanford University.

Rousseau, D. and Moulin, B. (1997). Mixed initiative in interactions between software agents, *Proceedings of the AAAI Spring Symposium*, pp. 135–137.

Roy, A. (2012). A theory of the brain: Localist representation is used widely in the brain, *Frontiers in Psychology* **3**: 551.

Rubin, D. C. and Wenzel, A. E. (1996). One hundred years of forgetting: A quantitative description of retention, *Psychological Review* **103**(4): 734–760.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). Learning representations by back-propagating errors, *Nature* **323**: 533–536.

Rumelhart, D. E., McClelland, J. L., and Group, P. R. (1986b). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision* **115**(3): 211–252.

Russell, J. A. and Barrett, L. F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant, *Journal of Personality and Social Psychology* **76**(5): 805–819.

Russell, S. and Wefald, E. (1988). Decision-theoretic control of reasoning: General theory and an application to game-playing, *Technical report*, University of California, Berkeley.

Russell, S. J. (1991). An architecture for bounded rationality, *ACM SIGART Bulletin* **2**(4): 146–150.

Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*, 4th edn, Pearson Education.

Russell, S. J. and Wefald, E. (1989). On optimal game-tree search using rational meta-reasoning, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 334–340.

Ruzzoli, M., Torralba, M., Fernández, L. M., and Soto-Faraco, S. (2019). The relevance of alpha phase in human perception, *Cortex* **120**: 249–268.

Ryder, J. M. and Zachary, W. W. (1991). Experimental validation of the attention switching component of the COGNET framework, *Proceedings of the Human Factors Society Annual Meeting*, Vol. 35, Sage Publications, pp. 72–76.

Ryle, G. (1945). Knowing how and knowing that: The presidential address, *Proceedings of the Aristotelian Society*, Vol. 46, pp. 1–16.

Sadowski, J. (2023). `https://twitter.com/jathansadowski/status/1625245803211272194`. Acessed: 13-05-2024.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. (2021). WinoGrande: An adversarial Winograd schema challenge at scale, *Communications of the ACM* **64**(9): 99–106.

Salgado, R., Prieto, A., Bellas, F., Calvo-Varela, L., and Duro, R. (2016). Motivational engine with autonomous sub-goal identification for the Multilevel Darwinist Brain, *Biologically Inspired Cognitive Architectures* **17**: 1–11.

Salvucci, D. D. (2000). A model of eye movements and visual attention, *Proceedings of the International Conference on Cognitive Modeling*, pp. 252–259.

Salvucci, D. D. (2002). Modeling driver distraction from cognitive tasks, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Salvucci, D. D. and Lee, F. J. (2003). Simple cognitive modeling in a complex cognitive architecture, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 265–272.

Salvucci, D. D., Chavez, A. K., and Lee, F. J. (2004). Modeling effects of age in complex tasks: A case study in driving, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1197–1202.

Samsonovich, A. V. (2010). Toward a unified catalog of implemented cognitive architectures, *Proceedings of the Annual Meeting of the BICA Society*, pp. 195–244.

Samsonovich, A. V. (2012). On a roadmap for the BICA Challenge, *Biologically Inspired Cognitive Architectures* **1**: 100–107.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II—Recent progress, *IBM Journal of Research and Development* **11**(6): 601–617.

Sandberg, A. and Bostrom, N. (2008). Whole brain emulation: A roadmap, *Technical Report 2008-3*, Oxford University.

Sanghi, P. and Dowe, D. L. (2003). A computer program capable of passing IQ tests, *Proceedings of the International Conference on Cognitive Science*, pp. 570–575.

Sanner, S., R Anderson, J., Lebiere, C., and Lovett, M. (2000). Achieving efficient and cognitively plausible learning in backgammon, *Proceedings of the International Conference on Machine Learning*, pp. 823–830.

Sanner, S. P. (1999). A quick introduction to 4CAPS programming, *Technical report*, Carnegie Mellon University.

Sarker, M. K., Zhou, L., Eberhart, A., and Hitzler, P. (2021). Neuro-symbolic artificial intelligence: Current trends, *arXiv:2105.05330*.

Saxe, A., Nelli, S., and Summerfield, C. (2021). If deep learning is the answer, what is the question?, *Nature Reviews Neuroscience* **22**(1): 55–67.

Scassellati, B. (1998). Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot, *Proceedings of the International Workshop on Computation for Metaphors, Analogy, and Agents*, pp. 176–195.

Scassellati, B. (2003). Investigating models of social development using a humanoid robot, *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 2704–2709.

Scellier, B. and Bengio, Y. (2017). Equilibrium propagation: Bridging the gap between energy-based models and backpropagation, *Frontiers in Computational Neuroscience* **11**: 24.

Schaeffer, R., Khona, M., and Fiete, I. (2022). No free lunch from deep learning in neuroscience: A case study through models of the entorhinal-hippocampal circuit, *Advances in Neural Information Processing Systems*, pp. 16052–16067.

Schaeffer, R., Miranda, B., and Koyejo, S. (2024). Are emergent abilities of Large Language Models a mirage?, *Advances in Neural Information Processing Systems* pp. 55565–55581.

Scharei, K., Heidecker, F., and Bieshaar, M. (2020). Knowledge representations in technical systems—A taxonomy, *arXiv:2001.04835*.

Scherer, K. (1994). Evidence for both universality and cultural spcificity of emotion elicitation, *in* P. Ekman and R. Davison (eds), *The Nature of Emotions: Fundamental Questions*, Oxford University Press, pp. 172–175.

Schermerhorn, P. W., Kramer, J. F., Middendorff, C., and Scheutz, M. (2006). DIARC: A testbed for natural human-robot interaction, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 45–52.

Schervish, M. J. (1996). P values: What they are and what they are not, *The American Statistician* **50**(3): 203–206.

Scheutz, M., Kramer, J., Middendorff, C., Schermerhorn, P., Heilman, M., Anderson, D., and Bui, P. (2005). Toward affective cognitive robots for human-robot interaction, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1737–1738.

Scheutz, M., Schermerhorn, P., Kramer, J., and Anderson, D. (2007). First steps toward natural human-like HRI, *Autonomous Robots* **22**(4): 411–423.

Scheutz, M., Harris, J., and Schermerhorn, P. (2013). Systematic integration of cognitive and robotic architectures, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, pp. 277–296.

Scheutz, M., Krause, E., and Sadeghi, S. (2014). An embodied real-time model of language-guided incremental visual search, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 1365–1370.

Scheutz, M., Williams, T., Krause, E., Oosterveld, B., Sarathy, V., and Frasca, T. (2019). An overview of the distributed integrated cognition affect and reflection DIARC architecture, *in* M. I. Aldinhas Ferreira, J. Silva Sequeira, and R. Ventura (eds), *Cognitive Architectures*, Springer, pp. 165–193.

Schiller, M. R. and Gobet, F. R. (2012). A comparison between cognitive and AI models of blackjack strategy learning, *in* B. Glimm and A. Krüger (eds), *KI 2012: Advances in Artificial Intelligence*, Springer, pp. 143–155.

Schlenoff, C., Madhavan, R., Albus, J., Messina, E., Barbera, T., and Balakirsky, S. (2005). Fusing disparate information within the 4D/RCS architecture, *Proceedings of the IEEE International Conference on Information Fusion*, pp. 1123–1130.

Schmidhuber, J. (2013). My first deep learning system of 1991+ deep learning timeline 1962–2013, *arXiv:1312.5548*.

Schmidt, F. L. and Oh, I.-S. (2016). The crisis of confidence in research findings in psychology: Is lack of replication the real problem? Or is it something else?, *Archives of Scientific Psychology* **4**(1): 32–37.

Schmidt, S. (2009). Shall we really do it again? The powerful concept of replication is neglected in the social sciences, *Review of General Psychology* **13**(2): 90–100.

Schneider, D. W. and Anderson, J. R. (2010). Asymmetric switch costs as sequential difficulty effects, *Quarterly Journal of Experimental Psychology* **63**(10): 1873–1894.

Schneider, D. W. and Anderson, J. R. (2011). A memory-based model of Hick's law, *Cognitive Psychology* **62**(3): 193–222.

Schneider, H. (2018). Meaningful-based cognitive architecture, *Procedia Computer Science* **145**: 471–480.

Schneider, H. (2019). Subsymbolic versus symbolic data flow in the Meaningful-Based Cognitive Architecture, *Proceedings of the Annual Meeting of the BICA Society*, pp. 465–474.

Schneider, J. and McGrew, K. (2013). The Cattell-Horn-Carroll (CHC) model of intelligence v2.2: A visual tour and summary, *Technical report*, Institute for Applied Psychometrics (IAP).

Schreckenghost, D., Bonasso, P., Kortenkamp, D., and Ryan, D. (1998). Three tier architecture for controlling space life support systems, *Proceedings of the International Joint Symposia on Intelligence and Systems*, IEEE, pp. 195–201.

Schrimpf, M. et al. (2018). Brain-score: Which artificial neural network for object recognition is most brain-like?, *bioRxiv:10.1101/407007*.

Schröder, T., Stewart, T. C., and Thagard, P. (2014). Intention, emotion, and action: A neural theory based on semantic pointers, *Cognitive Science* **38**(5): 851–880.

Schuhmann, C. et al. (2022). LAION-5b: An open large-scale dataset for training next generation image-text models, *arXiv:2210.08402*.

Schultheis, H. (2009). Computational and explanatory power of cognitive architectures: The case of ACT-R, *International Conference on Cognitive Modeling*, pp. 384–389.

Schunn, C. D., Crowley, K., and Okada, T. (1998). The growth of multidisciplinarity in the cognitive science society, *Cognitive Science* **22**(1): 107–130.

Schwalbe, G. and Finzel, B. (2023). A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts, *Data Mining and Knowledge Discovery* pp. 1–59.

Schwartz, D. G. (1995). *Cooperating Heterogeneous Systems*, Springer.

Schweizer, P. (1998). The truly total Turing test, *Minds and Machines* **8**(2): 263–272.

Searle, J. R. (1980). Minds, brains, and programs, *Behavioral and Brain Sciences* **3**(3): 417–424.

Selfridge, O. G. (1959). Pandemonium: A paradigm for learning, *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pp. 511–529.

Seth, A. K., McKinstry, J. L., Edelman, G. M., and Krichmar, J. L. (2004a). Active sensing of visual and tactile stimuli by brain-based devices, *International Journal of Robotics and Automation* **19**(4): 222–238.

Seth, A. K., McKinstry, J. L., Edelman, G. M., and Krichmar, J. L. (2004b). Visual binding through reentrant connectivity and dynamic synchronization in a brain-based device, *Cerebral Cortex* **14**(11): 1185–1199.

Shafir, E. and LeBoeuf, R. A. (2002). Rationality, *Annual Review of Psychology* **53**(1): 491–517.

Shapira, N., Levy, M., Alavi, S. H., Zhou, X., Choi, Y., Goldberg, Y., Sap, M., and Shwartz, V. (2024). Clever hans or neural theory of mind? Stress testing social reasoning in large language models, *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2257–2273.

Shapiro, S. C. (2006). Knowledge representation, *Encyclopedia of Cognitive Science*, Wiley Online Library.

Shapiro, S. C. and Ismail, H. O. (2003). Anchoring in a grounded layered architecture with integrated reasoning, *Robotics and Autonomous Systems* **43**(2-3): 97–108.

Shapiro, S. C. and Kandefer, M. (2005). A SNePS approach to the Wumpus World agent or Cassie meets the Wumpus, *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Nonmonotonic Reasoning, Action, and Change*, pp. 96–103.

Shapiro, S. C., Rapaport, W. J., Kandefer, M., Johnson, F. L., and Goldfain, A. (2007). Metacognition in SNePS, *AI Magazine* **28**(1): 17–31.

Sharir, O., Peleg, B., and Shoham, Y. (2020). The cost of training NLP models: A concise overview, *arXiv:2004.08900*.

Shastri, L. (1990). Connectionism and the computational effectiveness of reasoning, *Theoretical Linguistics* **16**(1): 65–87.

Shastri, L. (1999). Advances in SHRUTI—A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony, *Applied Intelligence* **11**(1): 79–108.

Shastri, L. (2001). From transient patterns to persistent structure: A model of episodic memory formation via cortico-hippocampal interactions, `https://www1.icsi.berkeley.edu/pubs/ai/transientpatterns04.pdf`.

Shastri, L. and Ajjanagadde, V. (1993). From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony, *Behavioral and Brain Sciences* **16**(3): 417–494.

Shepard, R. N., Hovland, C. I., and Jenkins, H. M. (1961). Learning and memorization of classifications, *Psychological Monographs: General and Applied* **75**(13): 1.

Shieber, S. M. (1994). Lessons from a restricted Turing test, *Communications of the Association for Computing Machinery* **37**(6): 70–78.

Shih, R., Dubrowski, A., and Carnahan, H. (2009). Evidence for haptic memory, *Proceedings of the Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 145–149.

Shimojo, S. and Shams, L. (2001). Sensory modalities are not separate modalities: Plasticity and interactions, *Current Opinion in Neurobiology* **11**(4): 505–509.

Shin, H. (1999). *Research Interactivity of Cognitive Science: A Bibliometric Analysis of Interdisciplinarity*, PhD thesis, The University of Texas at Austin.

Shneiderman, B. (1995). Looking for the bright side of user interface agents, *Interactions* **2**(1): 13–15.

Shoemaker, C. M. and Bornstein, J. A. (1998). The Demo III UGV program: A testbed for autonomous navigation research, *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 644–651.

Shoham, Y. (1999). What we talk about when we talk about software agents, *IEEE Intelligent Systems and Their Applications* **14**(2): 28–31.

Shortliffe, E. H., Axline, S. G., Buchanan, B. G., Merigan, T. C., and Cohen, S. N. (1973). An artificial intelligence program to advise physicians regarding antimicrobial therapy, *Computers and Biomedical Research* **6**(6): 544–560.

Shu, H. and Zhu, H. (2019). Sensitivity analysis of deep neural networks, *Proceedings of the Conference on Artificial Intelligence*, pp. 4943–4950.

Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., and Anderson, R. (2023). The curse of recursion: Training on generated data makes models forget, *arXiv:2305.17493*.

Silver, D. et al. (2016). Mastering the game of Go with deep neural networks and tree search, *Nature* **529**(7587): 484–489.

Silver, D. et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play, *Science* **362**(6419): 1140–1144.

Simmons, J. P., Nelson, L. D., and Simonsohn, U. (2016). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant, *Psychological Science* **22**(11): 1359–1366.

Simmons, R. (1989). Experience with a task control architecture for mobile robots, *Technical Report CMU-RI-TR-89-29*, Carnegie Mellon University.

Simmons, R. (1995). Towards reliable autonomous agents, *Proceedings of the AAAI Spring Symposium*, pp. 196–202.

Simmons, R., Krotkov, E., Whittaker, W., Albrecht, B., Bares, J., Fedor, C., Hoffman, R., Pangels, H., and Wettergreen, D. (1992). Progress towards robotic exploration of extreme terrain, *Journal of Applied Intelligence* **2**(2): 163–180.

Simmons, R., Fernandez, J., Goodwin, R., Koenig, S., and O'Sullivan, J. (2002). Xavier: An autonomous mobile robot on the web, *in* K. Goldberg and R. Siegwart (eds), *Beyond Webcams: An Introduction to Online Robots*, MIT Press, pp. 81–98.

Simmons, R. G. (1994a). Becoming increasingly reliable., *Proceedings of the Conference on Artificial Intelligence Planning Systems*, pp. 152–157.

Simmons, R. G. (1994b). Structured control for autonomous robots, *IEEE Transactions on Robotics and Automation* **10**(1): 34–43.

Simon, H. A. (1956). Rational choice and the structure of the environment, *Psychological Review* **63**(2): 129–138.

Simon, H. A. (1992). What is an "explanation" of behavior?, *Psychological Science* **3**(3): 150–161.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv:1409.1556*.

Siu, C. R. and Murphy, K. M. (2018). The development of human visual cortex and clinical implications, *Eye and Brain* pp. 25–36.

Skinner, B. (1957). *Verbal Behavior*, Copley Publishing Group.

Skinner, B. F. (1977). Why I am not a cognitive psychologist, *Behaviorism* **5**(2): 1–10.

Skorka, O. and Joseph, D. (2011). Toward a digital camera to rival the human eye, *Journal of Electronic Imaging* **20**(3): 033009.

Sloman, A. (2002). How many separately evolved emotional beasties live within us, *in* R. Trappl, P. Petta, and S. Payr (eds), *Emotions in Humans and Artifacts*, MIT Press, pp. 35–114.

Smith, C. A. and Kirby, L. D. (2001). Toward delivering on the promise of appraisal theory, *in* K. R. Scherer, A. Schorr, and T. Johnstone (eds), *Appraisal Processes in Emotion: Theory, Methods, Research*, Oxford University Press, pp. 121–138.

Smith, R. L., Lane, P. C., and Gobet, F. (2008). Modelling the relationship between visual short-term memory capacity and recall ability, *Proceedings of the European Symposium on Computer Modeling and Simulation*, pp. 99–104.

Smith, R. L., Gobet, F., and Lane, P. C. (2009). Checking chess checks with chunks: A model of simple check detection, *Proceedings of the International Conference on Cognitive modeling*.

Smith, S. D., Escobedo, R., Anderson, M., and Caudell, T. P. (1997). A deployed engineering design retrieval system using neural networks, *IEEE Transactions on Neural Networks* **8**(4): 847–851.

Solomonoff, R. (1956). Ray's notes on his thinking machine ideas, `https://raysolomonoff.com/dartmouth/boxbdart/boxbdart.html`.

Sowa, J. F. (1999). *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks/Cole Publishing Co.

Sowa, J. F. (2006). Semantic networks, *Encyclopedia of Cognitive Science*, Wiley Online Library.

Sowa, J. F. and Majumdar, A. K. (2003). Analogical reasoning, *Proceedings of the International Conference on Conceptual Structures*, pp. 16–36.

Spector, L. and Hendler, J. A. (1994). The use of supervenience in dynamic-world planning, *Proceedings of the Artificial Intelligence Planning Systems Conference*, pp. 158–163.

Sperling, G. (1960). The information available in brief visual presentations, *Psychological Monographs: General and Applied* **74**(11): 1.

Squire, L. R. (1992). Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory, *Journal of Cognitive Neuroscience* **4**(3): 232–243.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective, *Neurobiology of Learning and Memory* **82**(3): 171–177.

Squire, L. R. and Zola, S. M. (1988). Memory: Brain systems and behavior, *Trends in Neurosciences* **11**(4): 170–175.

Srivastava, A. et al. (2023). Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models, *Transactions on Machine Learning Research*.

Staat, W. (1993). On abduction, deduction, induction and the categories, *Transactions of the Charles S. Peirce Society* **29**(2): 225–237.

Stan, H. and Franklin, S. (2000). A behaviour instantiation agent architecture, *Connection Science* **12**(1): 21–44.

Stanovich, K. E. (2012). On the distinction between rationality and intelligence: Implications for understanding individual differences in reasoning, *in* K. J. Holyoak and R. G. Morrison (eds), *The Oxford Handbook of Thinking and Reasoning*, Oxford University Press, pp. 433–455.

Stanton, N. A., Eriksson, A., Banks, V. A., and Hancock, P. A. (2020). Turing in the driver's seat: Can people distinguish between automated and manually driven vehicles?, *Human Factors and Ergonomics in Manufacturing & Service Industries* **30**(6): 418–425.

Steels, L., Belpaeme, T., et al. (2005). Coordinating perceptually grounded categories through language: A case study for colour, *Behavioral and Brain Sciences* **28**(4): 469–488.

Stein, E. (1997). Can we be justified in believing that humans are irrational?, *Philosophy and Phenomenological Research* **27**(3): 545–565.

Sterrett, S. G. (2000). Turing's two tests for intelligence, *Minds and Machines* **10**(4): 541–559.

Stevens, J. R. (2017). Replicability and reproducibility in comparative psychology, *Frontiers in Psychology* **8**: 862.

Stevenson, R. J. (2010). An initial evaluation of the functions of human olfaction, *Chemical Senses* **35**(1): 3–20.

Stewart, T. and Eliasmith, C. (2012). Compositionality and biologically plausible models, *The Oxford Handbook of Compositionality*, Oxford University Press, pp. 595–615.

Stewart, T. and Eliasmith, C. (2013). Parsing sequentially presented commands in a large-scale biologically realistic brain model, *Proceedings of the Annual Meeting of the Cognitive Science Society*, pp. 3460–3467.

Stewart, T. C. and Eliasmith, C. (2009). Spiking neurons and central executive control: The origin of the 50-millisecond cognitive cycle, *Proceedings of the International Conference on Cognitive Modelling*, pp. 127–130.

Stokes, D. and Biggs, S. (2014). The dominance of the visual, *in* D. Stokes, M. Matthen, and S. Biggs (eds), *Perception and Its Modalities*, Oxford University Press, pp. 350–378.

Storrs, K. R. and Kriegeskorte, N. (2020). Deep learning for cognitive neuroscience, *in* D. Poeppel, G. R. Mangun, and M. S. Gazzaniga (eds), *The Cognitive Neurosciences*, MIT Press.

Strasburger, H. (2020). Seven myths on crowding and peripheral vision, *i-Perception* **11**(3): 2041669520913052.

Stroupe, A., Okon, A., Robinson, M., Huntsberger, T., Aghazarian, H., and Baumgartner, E. (2006). Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance, *Autonomous Robots* **20**: 113–123.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP, *arXiv:1906.02243*.

Su, N. M. and Crandall, D. J. (2021). The affective growth of computer vision, *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 9291–9300.

Suchman, L. A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge University Press.

Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era, *Proceedings of the International Conference on Computer Vision*, pp. 843–852.

Sun, R. (2003). A tutorial on CLARION 5.0, *Technical report*.

Sun, R. (2004). Desiderata for cognitive architectures, *Philosophical Psychology* **17**(3): 341–373.

Sun, R. (2007). The challenges of building computational cognitive architectures, *in* W. Duch and J. Mańdziuk (eds), *Challenges for Computational Intelligence*, Springer, pp. 37–60.

Sun, R. (2020). Potential of full human–machine symbiosis through truly intelligent cognitive systems, *AI & Society* **35**: 17–28.

Sun, R. and Fleischer, P. (2012). A cognitive social simulation of tribal survival strategies: The importance of cognitive and motivational factors, *Journal of Cognition and Culture* **12**(3-4): 287–321.

Sun, R. and Helie, S. (2013). Psychologically realistic cognitive agents: Taking human cognition seriously, *Journal of Experimental & Theoretical Artificial Intelligence* **25**(1): 65–92.

Sun, R. and Ling, C. X. (1998). Computational cognitive modeling, the source of power, and other related issues, *AI Magazine* **19**(2): 113–113.

Sun, R. and Peterson, T. (1998a). Hybrid learning incorporating neural and symbolic processes, *Proceedings of the International Conference on Fuzzy Systems*, Vol. 1, IEEE, pp. 727–732.

Sun, R. and Peterson, T. (1998b). Some experiments with a hybrid model for learning sequential decision making, *Information Sciences* **111**(1–4): 83–107.

Sun, R. and Wermter, S. (2000). *Hybrid Neural Systems*, Springer.

Sun, R., Merrill, E., and Peterson, T. (1998). A bottom-up model of skill learning, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 1037–1042.

Sun, R., Coward, L. A., and Zenzen, M. J. (2005). On levels of cognitive modeling, *Philosophical Psychology* **18**(5): 613–637.

Sun, R., Zhang, X., Slusarz, P., and Mathews, R. (2007). The interaction of implicit learning, explicit hypothesis testing learning and implicit-to-explicit knowledge extraction, *Neural Networks* **20**(1): 34–47.

Sun, R., Wilson, N., and Lynch, M. (2016). Emotion: A unified mechanistic interpretation from a cognitive architecture, *Cognitive Computation* **8**(1): 1–14.

Sutcliffe, G. and Suttner, C. (1998). The TPTP problem library, *Journal of Automated Reasoning* **21**: 177–203.

Sutton, R. S. (1992). Introduction: The challenge of reinforcement learning, *in* R. S. Sutton (ed.), *Reinforcement Learning*, Springer, pp. 1–3.

Sutton, R. S. and Barto, A. G. (1987). A temporal-difference model of classical conditioning, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 355–378.

Sweeney, L. (2003). That's AI?: A history and critique of the field, *Technical Report CMU-CS-03-106*, Carnegie Mellon University. `http://reports-archive.adm.cs.cmu.edu/anon/anon/home/ftp/2003/CMU-CS-03-106.pdf`.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks, *Proceedings of International Conference on Learning Representations*.

Szolovits, P. (1989). Expert systems tools and techniques: Past, present and future, *in* W. E. L. Grimson and R. S. Patil (eds), *AI in the 1980s and Beyond*, MIT Press, pp. 43–74.

Taatgen, N. A. (2020). A model of learning task-specific knowledge for a new task, *Proceedings of the Annual Conference of the Cognitive Science Society*, pp. 730–735.

Talamadupula, K., Benton, J., Kambhampati, S., Schermerhorn, P., and Scheutz, M. (2010). Planning for human-robot teaming in open worlds, *ACM Transactions on Intelligent Systems and Technology* **1**(2): 1–24.

Tan, A.-H., Lu, N., and Xiao, D. (2008). Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback, *IEEE Transactions on Neural Networks* **19**(2): 230–244.

Tanaka, H., Nayebi, A., Maheswaranathan, N., McIntosh, L., Baccus, S., and Ganguli, S. (2019). From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction, *Advances in Neural Information Processing Systems*, pp. 8537–8547.

Tansel, I. N., Mekdeci, C., and Mclaughlin, C. (1995). Detection of tool failure in end milling with wavelet transformations and neural networks (WT-NN), *International Journal of Machine Tools and Manufacture* **35**(8): 1137–1147.

Taylor, E. G. and Hummel, J. E. (2007). Perspectives on similarity from the LISA model, *Analogies: Integrating Multiple Cognitive Abilities* **5**: 21–25.

Taylor, J. E., Cortese, A., Barron, H. C., Pan, X., Sakagami, M., and Zeithamova, D. (2021). How do we generalize?, *Neurons, Behavior, Data Analysis and Theory* **1**: 001c.27687.

Taylor, R. (2011). Vision of beauty, *Physics World* **24**(5): 22.

Tecuci, G. (1991). A multistrategy learning approach to domain modeling and knowledge acquisition, *European Working Session on Learning*, Springer, pp. 14–32.

Tecuci, G. (1995). Building knowledge bases through multistrategy learning and knowledge acquisition, *in* G. Tecuci and Y. Kodratoff (eds), *Machine Learning and Knowledge Acquisition: Integrated Approaches*, Academic Press, pp. 13–50.

Tecuci, G. and Hieb, M. R. (1996). Teaching intelligent agents: The Disciple approach, *International Journal of Human-Computer Interaction* **8**(3): 259–285.

Tecuci, G., Boicu, M., Marcu, D., Stanescu, B., Boicu, C., and Barbulescu, M. (2004). Parallel knowledge base development by subject matter experts, *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, pp. 265–279.

Tecuci, G., Boicu, M., Boicu, C., Marcu, D., Stanescu, B., and Barbulescu, M. (2005). The Disciple–RKF learning and reasoning agent, *Computational Intelligence* **21**(4): 462–479.

Tecuci, G., Meckl, S., Marcu, D., and Boicu, M. (2019). Instructable cognitive agents for autonomous evidence-based reasoning, *Proceedings of the Annual Conference on Advances in Cognitive Systems*, pp. 73–92.

Tedersoo, L. et al. (2021). Data sharing practices and data availability upon request differ across scientific disciplines, *Scientific Data* **8**(192): 1–11.

Tesauro, G. (1991). Practical issues in temporal difference learning, *Advances in Neural Information Processing Systems*, pp. 259–266.

Thibadeau, R., Just, M. A., and Carpenter, P. A. (1982). A model of the time course and content of reading, *Cognitive Science* **6**(2): 157–203.

Thompson, J. A. (2021). Forms of explanation and understanding for neuroscience and artificial intelligence, *Journal of Neurophysiology* pp. 1860–1874.

Thórisson, K. and Helgasson, H. (2012). Cognitive architectures and autonomy: A comparative review, *Journal of Artificial General Intelligence* **3**(2): 1–30.

Thórisson, K. R. (1997). Layered modular action control for communicative humanoids, *Proceedings of Computer Animation*, pp. 134–143.

Thórisson, K. R. (1998). Real-time decision making in multimodal face-to-face communication, *Proceedings of the Second International Conference on Autonomous Agents*, pp. 16–23.

Thórisson, K. R. (1999). Mind model for multimodal communicative creatures and humanoids, *Applied Artificial Intelligence* **13**(4-5): 449–486.

Thórisson, K. R. (2002). Natural turn-taking needs no manual: Computational theory and model, from perception to action, *in* B. Granström, D. House, and I. Karlsson (eds), *Multimodality in language and speech systems*, Springer, pp. 173–207.

Tinbergen, N. (1951). *The Study of Instinct*, Oxford University Press.

Togelius, J., Yannakakis, G. N., Karakovskiy, S., and Shaker, N. (2013). Assessing believability, *in* P. Hingston (ed.), *Believable Bots*, Springer, pp. 215–230.

Toromanoff, M., Wirbel, E., and Moutarde, F. (2019). Is deep reinforcement learning really superhuman on Atari? Leveling the playing field, *arXiv:1908.04683*.

Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E., and Schultz, A. C. (2005). Enabling effective human-robot interaction using perspective-taking in robots, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **35**(4): 460–470.

Trafton, J. G., Fransen, B., Harrison, A. M., and Bugajska, M. (2009). An embodied model of infant gaze-following, *Technical Report ADA501497*, Naval Research Lab.

Trafton, J. G., Hiatt, L. M., Harrison, A. M., Tamborello, F. P., Khemlani, S. S., and Schultz, A. C. (2013). ACT-R/E: An embodied cognitive architecture for human-robot interaction, *Journal of Human-Robot Interaction* **2**(1): 30–55.

Traiger, S. (2000). Making the right identification in the Turing test, *Mind and Machines* **10**: 561–572.

Tsotsos, J. K. (1990). Analyzing vision at the complexity level, *Behavioral and Brain Sciences* **13**(3): 423–445.

Tsotsos, J. K. (1992). Image understanding, *in* S. C. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, 2nds edn, John Wiley & Sons, Inc., pp. 641–663.

Tsotsos, J. K. (2011). *A Computational Perspective on Visual Attention*, MIT Press.

Tsotsos, J. K. (2017). Complexity level analysis revisited: What can 30 years of hindsight tell us about how the brain might represent visual information?, *Frontiers in Psychology* **8**: 1216.

Tsotsos, J. K. and Kruijne, W. (2014). Cognitive programs: Software for attention's executive, *Frontiers in Psychology* **5**: 1260.

Tsotsos, J. K. and Luo, J. (2021). Probing the effect of selection bias on NN generalization with a thought experiment, *arXiv:2105.09934*.

Tsotsos, J. K., Mylopoulos, J., Cowey, H. D., and Zucker, S. W. (1980). A framework for visual motion understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**(6): 563–573.

Tsotsos, J. K., Culhane, S. M., Wai, W. Y. K., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling visual attention via selective tuning, *Artificial Intelligence* **78**(1-2): 507–545.

Tsotsos, J. K., Kotseruba, I., Rasouli, A., and Solbach, M. D. (2018). Visual attention and its intimate links to spatial cognition, *Cognitive Processing* **19**: 121–130.

Tuckute, G., Feather, J., Boebinger, D., and McDermott, J. H. (2022). Many but not all deep neural network audio models capture brain responses and exhibit hierarchical region correspondence, *bioRxiv:/2022.09.06.506680*.

Tulving, E. (1972). Episodic and semantic memory, *in* E. Tulving and W. Donaldwon (eds), *Organization of Memory*, Academic Press, pp. 381–403.

Tulving, E. et al. (2002). Episodic memory: From mind to brain, *Annual Review of Psychology* **53**(1): 1–25.

Turing, A. (1937). On computable numbers, with an application to the entscheidungsproblem, *Proceedings of the London Mathematical Society*, Vol. s2-42, pp. 230–265.

Turing, A. (1950). Computing machinery and intelligence, *Mind* **59**(236): 433–460.

Tyler, S. W., Neukom, C., Logan, M., and Shively, J. (1998). The MIDAS human performance model, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pp. 320–324.

Tyrrell, T. (1993). *Computational Mechanisms for Action Selection*, PhD thesis, University of Edinburgh.

Tyrrell, T. (1994). An evaluation of Maes's bottom-up mechanism for behavior selection, *Adaptive Behavior* **2**(4): 307–348.

Udandarao, V., Prabhu, A., Ghosh, A., Sharma, Y., Torr, P. H., Bibi, A., Albanie, S., and Bethge, M. (2024). No "zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance, *Proceedings of the International Conference on Learning Representations Workshop on Data Problems for Foundation Models*.

Ullman, S. (1983). Visual routines, *Technical Report 723*, MIT.

Ultsch, A. (1998). The integration of connectionist models with knowledge-based systems: Hybrid systems, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1530–1535.

Ulutas, B., Erdemir, E., and Kawamura, K. (2008). Application of a hybrid controller with non-contact impedance to a humanoid robot, *Proceedings of the IEEE International Workshop on Variable Structure Systems*, pp. 378–383.

Valmeekam, K., Marquez, M., Sreedharan, S., and Kambhampati, S. (2023). On the planning abilities of large language models-a critical investigation, *Advances in Neural Information Processing Systems* **36**: 75993–76005.

Van Rooij, I. (2008). The tractable cognition thesis, *Cognitive Science* **32**(6): 939–984.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer.

Varchavskaia, P., Fitzpatrick, P., and Breazeal, C. (2001). Characterizing and processing robot-directed speech, *Proceedings of the International Conference on Humanoid Robotics*, Vol. 488.

Varma, S. (2014a). The CAPS family of cognitive architectures, *in* S. E. F. Chipman (ed.), *The Oxford Handbook of Cognitive Science*, pp. 49–68.

Varma, S. (2014b). The subjective meaning of cognitive architecture: A Marrian analysis, *Frontiers in Psychology* **5**: 1–9.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need, *Advances in Neural Information Processing Systems* **30**: 5998–6008.

Vaux, J. (2001). From expert systems to knowledge-based companies: How the ai industry negotiated a market for knowledge, *Social Epistemology* **15**(3): 231–245.

Veksler, V. D., Hoffman, B. E., and Buchler, N. (2022). Symbolic deep networks: A psychologically inspired lightweight and efficient approach to deep learning, *Topics in Cognitive Science* **14**(4): 702–717.

Veloso, M., Carbonell, J., Perez, A., Borrajo, D., Fink, E., and Blythe, J. (1995). Integrating planning and learning: The PRODIGY architecture, *Journal of Experimental & Theoretical Artificial Intelligence* **7**(1): 81–120.

Veloso, M. M. (1994). PRODIGY/ANALOGY: Analogical reasoning in general problem solving, *in* M. Richer, S. Wess, K. Althoff, and F. Maurer (eds), *Topics in Case-Based Reasoning*, pp. 33–52.

Veloso, M. M. and Carbonell, J. G. (1993). Toward scaling up machine learning: A case study with derivational analogy in PRODIGY, *in* S. Minton (ed.), *Machine Learning Methods for Planning*, Elsevier, pp. 233–272.

Vernon, D. (2014). *Artificial Cognitive Systems: A Primer*, MIT Press.

Vernon, D. (2016). Two ways (not) to design a cognitive architecture, *Proceedings of the European Society for Cognitive Systems (EUCognition)*, pp. 42–43.

Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents, *IEEE Transactions on Evolutionary Computation* **11**(2): 151–180.

Vernon, D., von Hofsten, C., and Fadiga, L. (2016). Desiderata for developmental cognitive architectures, *Biologically Inspired Cognitive Architectures* **18**: 116–127.

Verschure, P. F. (2012). Distributed adaptive control: A theory of the mind, brain, body nexus, *Biologically Inspired Cognitive Architectures* **1**: 55–72.

Verschure, P. F. and Althaus, P. (2003). A real-world rational agent: Unifying old and new AI, *Cognitive Science* **27**(4): 561–590.

Verschure, P. F. and Voegtlin, T. (1998). A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: Distributed Adaptive Control III, *Neural Networks* **11**(7-8): 1531–1549.

Verschure, P. F., Voegtlin, T., and Douglas, R. J. (2003). Environmentally mediated synergy between perception and behaviour in mobile robots, *Nature* **425**(6958): 620–624.

Vigdor, B. and Lerner, B. (2007). The Bayesian ARTMAP, *IEEE Transactions on Neural Networks* **18**(6): 1628–1644.

Villalobos, P., Sevilla, J., Heim, L., Besiroglu, T., Hobbhahn, M., and Ho, A. (2024). Will we run out of data? An analysis of the limits of scaling datasets in Machine Learning, *Proceedings of the International Conference on Machine Learning*.

Vinokurov, Y., Lebiere, C., Szabados, A., Herd, S., and O'Reilly, R. (2013). Integrating top-down expectations with bottom-up perceptual processing in a hybrid neural-symbolic architecture, *Biologically Inspired Cognitive Architectures* **6**: 140–146.

Von Ahn, L., Blum, M., and Langford, J. (2004). Telling humans and computers apart automatically, *Communications of the ACM* **47**(2): 56–60.

von Neumann, J. (1993). First draft of a report on the EDVAC, *IEEE Annals of the History of Computing* **15**(4): 27–75.

Vosniadou, S. (1995). Analogical reasoning in cognitive development, *Metaphor and Symbol* **10**(4): 297–308.

Vouloutsi, V., Munoz, M. B., Grechuta, K., Lallee, S., Duff, A., Llobet, J.-Y. P., and Verschure, P. (2015). A new biomimetic approach towards educational robotics: The Distributed Adaptive Control of a synthetic tutor assistant, *Proceedings of the International Symposium on New Frontiers in Human-Robot Interaction*, pp. 22–29.

Waller, B. N. (2001). Classifying and analyzing analogies, *Informal Logic* **21**(3): 199–218.

Wang, J., Feng, K., and Wu, J. (2019). SVM-based deep stacking networks, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 5273–5280.

Wang, P. (1995a). *Non-Axiomatic Reasoning System—Exploring the Essence of Intelligence*, PhD thesis, Indiana University.

Wang, P. (1995b). On the working definition of intelligence, *Technical Report 94*, Indiana University.

Wang, P. (2006a). Artificial intelligence: What it is, and what it should be, *Proceedings of AAAI Spring Symposium*.

Wang, P. (2006b). *Rigid Flexibility: The Logic of Intelligence*, Springer.

Wang, P. (2009). Insufficient knowledge and resources—A biological constraint and its functional implications, *Proceedings of the AAAI Fall Symposium*.

Wang, P. (2012). Solving a problem with or without a program, *Journal of Artificial General Intelligence* **3**(3): 43.

Wang, P. (2013). Natural language processing by reasoning and learning, *International Conference on Artificial General Intelligence*, Springer, pp. 160–169.

Wang, P. (2022). Intelligence: From definition to design, *International Workshop on Self-Supervised Learning*, pp. 35–47.

Wang, P., Talanov, M., and Hammer, P. (2016). The emotional mechanisms in NARS, *Proceedings of the International Conference on Artificial General Intelligence*, Springer, pp. 150–159.

Wang, T. T., Gleave, A., Belrose, N., Tseng, T., Miller, J., Dennis, M. D., Duan, Y., Pogrebniak, V., Levine, S., and Russell, S. (2022). Adversarial policies beat professional-level Go AIs, *Advances on Neural Information Processing Systems Workshop on Deep Reinforcement Learning*.

Wang, X. and Carbonell, J. G. (1994). Learning by observation and practice: Towards real applications of planning systems, *Proceedings of the AAAI Fall Symposium*.

Wang, Y., Yao, Q., Kwok, J. T., and Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning, *ACM Computing Surveys* **53**(3): 1–34.

Warwick, K. and Shah, H. (2016). Can machines think? A report on Turing test experiments at the Royal Society, *Journal of experimental & Theoretical artificial Intelligence* **28**(6): 989–1007.

Wasserman, J. D. (2019). Deconstructing chc, *Applied Measurement in Education* **32**(3): 249–268.

Wasson, G., Kortenkamp, D., and Huber, E. (1999). Integrating active perception with an autonomous robot architecture, *Robotics and Autonomous Systems* **29**(2-3): 175–186.

Watkins, C. J. C. H. (1989). *Learning From Delayed Rewards*, PhD thesis, Cambridge University.

Watson, J. B. (1920). Is thinking merely the action of language mechanisms?, *British Journal of Psychology* **11**: 87–104.

Watt, S. (1996). Naive psychology and the inverted Turing test, *Psycoloquy* **7**(14): 463–518.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W. (2022). Emergent abilities of large language models, *Transactions on Machine Learning Research*.

Weintraub, J. (1992). History of the PC therapist, https://web.archive.org/web/20050107092226/http://www.loebner.net:80/Prizef/weintraub-bio.html.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine, *Communications of the ACM* **9**(1): 36–45.

Wendelken, C. and Shastri, L. (2002). Combining belief and utility in a structured connectionist agent architecture, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Wendelken, C. and Shastri, L. (2003). Acquisition of concepts and causal rules in SHRUTI, *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Weng, J. (2002). A theory for mentally developing robots, *Proceedings of the IEEE International Conference on Development and Learning*, pp. 131–140.

Weng, J. (2007). On developmental mental architectures, *Neurocomputing* **70**(13–15): 2303–2323.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University.

Werning, M. and Cheng, S. (2017). Taxonomy and unity of memory, *The Routledge Handbook of Philosophy of Memory*, Routledge, pp. 7–20.

White, K. G. (2001). Forgetting functions, *Animal Learning & Behavior* **29**(3): 193–207.

White, S., Chen, J., and Forsyth, B. (2010). Reading-related literacy activities of American adults: Time spent, task types, and cognitive skills used, *Journal of Literacy Research* **42**(3): 276–307.

Whittington, J. C. and Bogacz, R. (2019). Theories of error back-propagation in the brain, *Trends in Cognitive Sciences* **23**(3): 235–250.

Wickens, C., McCarley, J., and Thomas, L. (2003). Attention-situation awareness (A-SA) model, *Proceedings of the NASA Aviation Safety Program Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*, pp. 189–207.

Wiener, N. (1948). *Cybernetics or Control and Communication in the Animal and the Machine*, MIT Press.

Wiley, J. F. (2016). *R Deep Learning Essentials*, Packt Publishing.

Wilhelm, O. and Kyllonen, P. (2021). To predict the future, consider the past: Revisiting Carroll (1993) as a guide to the future of intelligence research, *Intelligence* **89**: 101585.

Williams, T., Matuszek, C., Mead, R., and Depalma, N. (2024). Scarecrows in Oz: The use of large language models in HRI, *ACM Transactions on Human-Robot Interaction* **13**(1): 1–11.

Wilson, I. (1979). Foundations of hierarchical control, *International Journal of Control* **29**(6): 899–933.

Wilson, J. R., Krause, E., Scheutz, M., and Rivers, M. (2016). Analogical generalization of actions from single exemplars in a robotic architecture, *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems*, pp. 1015–1023.

Wilson, N. R. (2012). *Towards a Psychologically Plausible Comprehensive Computational Theory of Emotion*, PhD thesis, Rensselaer Polytechnic Institute.

Wimmer, H. and Perner, J. (1983). Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception, *Cognition* **13**(1): 103–128.

Winne, P. H. and Azevedo, R. (2014). Metacognition, *in* R. K. Sawyer (ed.), *The Cambridge Handbook of the Learning Sciences*, Cambridge University Press, pp. 63–87.

Winograd, T. (1975). Frame representations and the declarative/procedural controversy, *in* D. G. Bobrow and A. Collins (eds), *Representation and Understanding*, Elsevier, pp. 185–210.

Winograd, T. (1980). What does it mean to understand language?, *Cognitive Science* **4**(3): 209–241.

Winston, P. H. (1992). *Artificial Intelligence*, 3rd edn, Addison-Wesley.

Wloka, C., Kotseruba, I., and Tsotsos, J. K. (2018). Active fixation control to predict saccade sequences, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3184–3193.

Wojciechowski, J. Q. (2004). Validation of improved research integration tool (IMPRINT) driving model for workload analysis, *Technical Report ARL-TR-3145*, Army Research Laboratory.

Wolf, M. T., Assad, C., Kuwata, Y., Howard, A., Aghazarian, H., Zhu, D., Lu, T., Trebi-Ollennu, A., and Huntsberger, T. (2010). 360-degree visual detection and target tracking on an autonomous surface vehicle, *Journal of Field Robotics* **27**(6): 819–833.

Wolfe, J. M. (1994). Guided Search 2.0: A revised model of visual search, *Psychonomic Bulletin & Review* **1**(2): 202–238.

Wong, C., Kortenkamp, D., and Speich, M. (1995). A mobile robot that recognizes people, *Proceedings of International Conference on Tools with Artificial Intelligence*, IEEE, pp. 346–353.

Wood, S. (1995). When being reactive just won't do, *Proceedings of the AAAI Spring Symposium on Integrated Planning Applications*, pp. 102–106.

Wooldridge, M. (1999). Intelligent agents, *in* G. Weiss (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, pp. 27–51.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice, *The Knowledge Engineering Review* **10**(2): 115–152.

Worden, R., Bennett, M. S., and Neacsu, V. (2021). The thalamus as a blackboard for perception and planning, *Frontiers in Behavioral Neuroscience* **15**: 633872.

Wray, R. E. and Chong, R. S. (2003). Quantitative explorations of category learning with symbolic concept acquisition, *Proceedings of the International Conference on Cognitive Modeling*.

Wray, R., Chong, R., Phillips, J., Rogers, S., and Walsh, B. (1992). A survey of cognitive and agent architectures, http://ai.eecs.umich.edu/cogarch0/.

Wurman, P. R. et al. (2022). Outracing champion gran turismo drivers with deep reinforcement learning, *Nature* **602**(7896): 223–228.

Wyatte, D., Herd, S., Mingus, B., and O'Reilly, R. (2012). The role of competitive inhibition and top-down feedback in binding during object recognition, *Frontiers in Psychology* **3**: 182.

Xiong, Z., Zhang, Y., Wu, F., and Zeng, W. (2017). Computational depth sensing: Toward high-performance commodity depth cameras, *IEEE Signal Processing Magazine* **34**(3): 55–68.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex, *Proceedings of the National Academy of Sciences* **111**(23): 8619–8624.

Young, M. J. (1993). Successively approximating human performance, *Technical Report AD-A272 186*, Armstrong Laboratory.

Zachary, W., Ryder, J. M., Ross, L. R., and Zubritzky, M. C. (1990). Validation and application of COGNET model of human-computer interaction in Naval Air ASW, *Technical Report 900531-8704*, CHI Systems.

Zachary, W., Santarelli, T., Ryder, J., and Stokes, J. (2000). Developing a multi-tasking cognitive agent using the COGNET/iGEN integrative architecture, *Technical Report ADA416891*, CHI Systems Inc.

Zachary, W. W., Ryder, J. M., and Hicinbothom, J. H. (1998). Cognitive task analysis and modeling of decision making in complex environments, *in* J. A. Cannon-Bowers and E. Salas (eds), *Making Decisions Under Stress: Implications for Individual and Team Training*, American Psychological Association, pp. 315–345.

Zador, A. M. (2019). A critique of pure learning and what artificial neural networks can learn from animal brains, *Nature Communications* **10**(1): 1–7.

Zall, R. and Kangavari, M. R. (2022). Comparative analytical survey on cognitive agents with emotional intelligence, *Cognitive Computation* **14**: 1223–1246.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks, *Proceedings of the European conference on Computer Vision*, pp. 818–833.

Zeng, H. (2022). What is a cell type and how to define it?, *Cell* **185**(15): 2739–2755.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization, *Communications of the ACM* **64**(3): 107–115.

Zhang, N., Weng, J., and Zhang, Z. (2002). A developing sensory mapping for robots, *Proceedings of the IEEE International Conference on Development and Learning*, pp. 13–20.

Zhang, Y. and Weng, J. (2002). Action chaining by a developmental robot with a value system, *Proceedings of the IEEE International Conference on Development and Learning*, pp. 53–60.

Zhang, Y., Weng, J., and Hwang, W.-S. (2005). Auditory learning: A developmental method, *IEEE Transactions on Neural Networks* **16**(3): 601–616.

Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: A survey, *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 737–744.

Zhao, Y., Wang, G., Tang, C., Luo, C., Zeng, W., and Zha, Z.-J. (2021). A battle of network structures: An empirical study of CNN, Transformer, and MLP, *arXiv:2108.13002*.

Zhou, B., Bau, D., Oliva, A., and Torralba, A. (2018). Interpreting deep visual representations via network dissection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(9): 2131–2145.

Zhou, Z.-H. and Feng, J. (2017). Deep forest: Towards an alternative to deep neural networks, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3553–3559.

Zimmermann, R. S., Borowski, J., Geirhos, R., Bethge, M., Wallis, T., and Brendel, W. (2021). How well do feature visualizations support causal understanding of CNN activations?, *Advances in Neural Information Processing Systems* **34**: 11730–11744.

Zmigrod, S. and Hommel, B. (2013). Feature integration across multimodal perception and action: A review, *Multisensory Research* **26**(1-2): 143–157.

Zoph, B. and Le, Q. V. (2017). Neural architecture search with reinforcement learning, *Proceedings of International Conference on Learning Representations*.

Index