Compositionality as a mechanism of intelligence in brains and machines

Aimen Zerroug

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *01/12/2023* **par :**

Aimen ZERROUG

Compositionality as a Mechanism of Intelligence in Brains and Machines

---

JURY

NICOLAS ROUGIER	Inria Bordeaux, France	Rapporteur
FRÉDÉRIC ALEXANDRE	Inria Bordeaux, France	Rapporteur
RUFIN VANRULLEN	CNRS Toulouse, France	Examinateur
JULIE HUNTER	LINAGORA Labs	Examinatrice
THOMAS SERRE	Brown University, USA	Directeur de These
NICHOLAS ASHER	ANITI, France	Co-Directeur de These

École doctorale et spécialité :

EDMITT : Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse – Informatique et Télécommunications

Unité de Recherche :

IRIT, Institut de Recherche en Informatique de Toulouse

Directeur(s) de Thèse :

Thomas SERRE et *Nicholas ASHER*

Rapporteurs :

Nicolas Rougier (Université de Bordeaux) et *Frédéric Alexandre (Université de Bordeaux)*

RÉSUMÉ

L'intelligence humaine est caractérisée par la flexibilité et la généralisation. La perception de nouveaux concepts par les humains et leur exécution de nouvelles tâches nécessitent souvent peu d'effort s'ils reconnaissent déjà des concepts et des tâches similaires. La compositionnalité a longtemps été considérée comme une caractéristique fondamentale de l'intelligence humaine sous-tendant ces capacités.

Contrairement aux humains, les modèles d'apprentissage profond qui atteignent des performances surhumaines dans de nombreuses tâches manquent de généralisation et s'adaptent mal aux changements de tâches. De plus, les modèles présentant des capacités de généralisation modestes nécessitent de grandes quantités d'expérience d'apprentissage. Ces mêmes modèles n'exploitent pas la compositionnalité dans l'apprentissage ou l'inférence. Cette thèse discute d'abord des caractéristiques générales de l'intelligence humaine, puis se plonge dans la littérature sur la compositionnalité en IA. Le résumé de ces résultats met en lumière des lacunes dans ces domaines ; un aspect peu exploré de la compositionnalité dans l'apprentissage automatique, à savoir le rôle de la compositionnalité dans l'apprentissage, pourrait être crucial pour la généralisation dans les modèles d'apprentissage profond.

L'importance de la compositionnalité dans l'apprentissage a motivé l'investigation de la capacité des modèles de réseaux neuronaux à décomposer les tâches en leurs composants élémentaires et à composer les compétences acquises pour résoudre de nouvelles tâches. En utilisant le raisonnement visuel comme une tâche de base, j'ai développé un test de raisonnement visuel qui évalue l'efficacité de l'échantillonnage et de l'apprentissage compositionnel des modèles de vision standard. Les expériences démontrent que même les modèles pré-entraînés nécessitent plus d'échantillons pour atteindre les performances humaines. De plus, même si les modèles sont capables de réutiliser les compétences apprises à partir de tâches élémentaires pour résoudre efficacement leurs compositions, ils ne décomposent pas les tâches apprises en leurs composants élémentaires lors de l'apprentissage.

Ces idées et ces résultats soulignent l'importance des stratégies d'apprentissage et de l'expérience dans la formation des systèmes d'apprentissage et leur impact sur l'intelligence. En conséquence, le dernier chapitre s'inspire de la fonction du cerveau et dérive des principes clés de conception de réseaux neuronaux pour faire avancer le domaine vers l'intelligence humaine. J'utilise ces principes pour proposer des méthodes de conception et d'entraînement de réseaux neuronaux

basées sur la modularité, l'agence, le contrôle du temps et des ressources de calcul et l'apprentissage programmé. De plus, je propose un prototype démontrant ces méthodes ; une architecture modulaire polyvalente avec un contrôle sur ses calculs internes nommée AbstractNet. Les expériences et l'analyse d'AbstractNet montrent sa capacité à effectuer et à apprendre des tâches hétérogènes et homogènes, à adapter le nombre de calculs aux exigences de la tâche et à utiliser des stratégies de routage diverses pour résoudre différentes tâches. En tant que preuve de concept, AbstractNet n'incorpore pas tous les principes de conception proposés. Il pourrait être enrichi de ces principes de conception pour acquérir des capacités de haut niveau telles qu'apprendre à apprendre et à concevoir et simuler un modèle de son environnement.

Dans le chapitre de conclusion, j'analyse les implications plus larges de cette recherche dans le contexte du débat en cours sur les principes computationnels essentiels à l'intelligence, tout en identifiant également des pistes potentielles pour des explorations futures.

ABSTRACT

Human intelligence is known for its flexibility and generalization. Humans can easily understand new concepts and perform new tasks if they have prior experience with similar concepts and tasks. Compositionality is considered to be a key feature of human intelligence that supports these capacities.

On the other hand, deep-learning models that perform better than humans in many tasks lack generalization and often fail to adapt to changes in tasks. Although some models show modest generalization capabilities, they require a large amount of learning experience. These models also do not take advantage of compositionality in learning or inference.

This dissertation discusses the general characteristics of human intelligence and explores the literature on compositionality in AI. The research identifies gaps in these fields, particularly in the role of compositionality in learning, which could be crucial for generalization in deep learning models.

The study investigates the capacity of neural network models to decompose tasks into their elementary components and use the learned skills to solve new tasks, using visual reasoning as a test case. The research develops a visual reasoning benchmark that evaluates the sample efficiency and compositional learning standards for vision models. The experiments demonstrate that even pre-trained models require significantly more samples to reach human performance. Additionally, the baseline models can reuse learned skills from elementary tasks to solve their compositions efficiently, but they do not decompose the learned tasks into their elementary components during learning.

These findings highlight the importance of learning strategies and experience in shaping learning systems and their impact on intelligence. The final chapter proposes a framework for neural network design and training based on modularity, agency, control over computational time and resources, and curriculum learning. The chapter also presents a proof of concept for the framework, a general-purpose modular architecture named AbstractNet. The experiments and analysis of AbstractNet show its capacity for multi-tasking several heterogeneous and homogeneous tasks, adapting the number of computations to task demands, and using various routing strategies for solving different tasks. The study identifies potential avenues for future exploration in the broader context of the ongoing discourse on the computational principles essential for intelligence.

ACKNOWLEDGMENTS

Throughout the process of writing this dissertation and embarking on the long path toward obtaining my PhD, I have been incredibly fortunate. I extend my heartfelt gratitude to the many individuals who have played diverse roles and contributed significantly to this transformative journey.

First, I would like to thank my advisor, Thomas Serre, for giving me this unique opportunity to research neuroscience and AI. He has provided me with ample support and guidance throughout this PhD, always asking the right questions and proposing the right experiments. His guidance always balanced supervision and freedom, through which I learned to build scientific ideas and develop critical thinking. His willingness to support my ideas and provide me with opportunities for collaboration gave me confidence and promoted my growth as a researcher. I will always be thankful to him for being patient with my mistakes and comprehensive during tough times. I have learned a lot from him, both on a professional and personal level.

I also thank Nicolas Asher for being my supervisor and the direction team of ANITI for giving me and my colleagues this opportunity and supporting the organization throughout these difficult years. Thanks to financial support from ANITI, I was able to visit Brown University as a short-term scholar, which contributed greatly to my research experience. I also sincerely thank Corinne Joffre for her immense help throughout the years.

I offer a special thanks to my colleagues and collaborators at ANITI. Our conversations shaped my thinking and developed into fascinating projects that formed the basis for this thesis. I also want to thank the other members of *Serre Lab*, Lakshmi N. Govindarajan, Rex Liu, Lore Goetschalckx, and Drew Linsley. They were very welcoming during my stay at Brown University as a visitor, and we shared interesting discussions and memorable moments.

During my visits, I have had the pleasure of being introduced to Sebastian Musslick, who has been a mentor and a great collaborator in the final years of my PhD. Discussions with him have shaped my thoughts about compositionality, and his experience in behavioral experiment design has been indispensable for our collaboration. I admire his benevolence and passion for research, and I sincerely thank him for his inspiration and help.

My special gratitude goes to my great friend, Mohit Vaishnav; he has been a great partner in research and a roommate during our travels to Brown University. I will always treasure the moments we spent together.

I offer great thanks to Rufin VanRullen, as he welcomed me into his research

team and offered me a place alongside his students. His deep insight into neuroscience and AI and sharp critical thinking always gave me a different perspective on my ideas. I will always be thankful for his kind support and optimism about my ideas. The days that I spent in CerCo with all my friends and colleagues were invaluable. These include Andrea Alamia, Milad Mozafari, Bhavin Choksi, Samson Chota, Javier Cuadrado, Colin Decourt, Anais Servais, and Ismael Khalfaoui (among others). My apologies to the many important people I have likely forgotten to mention. I had many fun moments while sharing food, laughs, and discussions on science. I will treasure the moments we spent together.

On a less academic note, I would like to thank my family. My parents, my brothers, and my sister have offered me mental support and encouraged me to pursue a PhD. I also thank many dear friends, Omar, Islem, Abderrahman, and Nassim, some of whom share my path towards a PhD, who have been a constant source of happiness for me throughout the years, even though we are separated by long distances.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The key to human intelligence is the ability to "make infinite use of finite means"

– Wilhelm von Humboldt

The ultimate goal of the AI field is to build machines that reach or surpass human intelligence. This goal dates back to the inception of the computer, where Alan Turing stated, "What we want is a machine that can learn from experience" in a lecture in 1947. Such a grand challenge has proven difficult, given the endless possibilities and the broadness of the goal. Indeed, the field has experienced decades of countless attempts and cycles of changing interest in research directions. Researchers resorted to chipping away at the problem by placing the goalpost on achievable objectives, such as recognizing digits to solve captchas or playing chess. The trend of simplification resulted in the field subdividing into several sub-fields and the development of highly specialized systems. Although this trend has resulted in a bias within the field to build specialized systems, this was an unavoidable step toward achieving the broader goal of human intelligence. Accordingly, during recent decades, the field of artificial intelligence has made significant strides in several tasks, and the highly specialized systems of the past have been replaced by gradually more general systems. Large Language Models [OpenAI, 2023], for example, have made substantial progress in Natural Language Processing as they can solve a variety of tasks with high accuracy. Such large-scale models [Bubeck et al., 2023, Ramesh et al., 2022] show great promise for reaching human-level intelligence.

Even though deep neural networks (DNNs) have made many developments in human intelligence, they still lag behind the brain in flexibility and efficiency. The brain learns numerous skills, giving humans the capacity to understand tasks from descriptions, use knowledge acquired by solving prior tasks, learn from scarce examples, and reflect on their behavior. On the other hand, deep learning models require large amounts of data to learn a task and generalize poorly to novel tasks or changes in input statistics without additional training. These shortcomings of DNNs show that they lack crucial components for reaching the level of human intelligence.

Naturally, several researchers have used the brain as inspiration for developing

more intelligent systems at many levels of granularity, from the spiking of neurons to the mechanisms of memory and attention to cognitive functions such as control and simulation. While the brain's true workings remain enigmatic, this approach relies on theories that attempt to explain its function. Nonetheless, it is a rational and compelling strategy since it has witnessed wide popularity in recent decades with varying degrees of success and failure. Furthermore, modern AI systems only surpass humans in accuracy and fail in terms of robustness, learning efficiency, and generalization. A promising path towards improving AI models in these aspects is to take inspiration from a system that excels at them.

Given the brain's overwhelming complexity and our limited understanding of its function, it is important to first identify the principles that characterize its intelligence, relate them to the mechanisms of its function, and then translate them into DNN design principles that guide their implementation. In this dissertation, I focus on compositionality as a key aspect of human intelligence. The principle of compositionality has been used for characterizing language and thought [Frege, 1980, Fodor, 1975] stating that the meaning of the whole is a function of its components and their structure. This principle appears in many disciplines of science, including deep learning, where it is used for designing neural architectures, evaluating models, and creating tasks. However, its application has not successfully closed the gap between brains and machines.

The general aim of this dissertation is to promote a direction of research that focuses on all aspects of human intelligence, with a special focus on compositionality, by proposing a framework for developing and training brain-inspired architectures. The first chapter positions my work within the broader field by exploring characterizations of human intelligence and then discussing the definition and manifestation of compositionality in the brain and deep learning models.

1.1 Characterizing Human Intelligence

Among the issues in the AI field is the lack of consensus on a definition of general intelligence. This results in disparities in tasks and evaluation methods for AI models. Although the field advances at a rapid pace, it lacks direction. A much-needed definition of intelligence should outline characterizations of intelligent systems. Given the clear characteristics of intelligence, model design could be guided towards clear goals and driven by comprehensive evaluation methods.

Several descriptions and measures have been proposed in recent times [Lake et al., 2016, Chollet, 2019] and focus on the capacity for skill acquisition and generalization rather than the capacity to learn individual tasks. Although these works have moderately influenced the AI field, performance remains the main driver of model design and metrics in AI benchmarks.

Interest in intelligence has emerged early in philosophical discussions on the nature of the mind. Our understanding of the concept has evolved throughout the decades, shaped by theories and developments in intelligence tests and psychometric studies. From Charles Spearman's G factor and Raymond Cattel's fluid and crystallized intelligence to ideas contributed to cognitive science and AI research, the definitions of intelligence have progressively become more general. Views on intelligence generally and human intelligence specifically have changed, especially since the start of AI research. The goal of many pioneers in the field of AI was to develop machines that emulate human intelligence, for example, by playing board games or conversing with humans without being detected as robots. The prime source of inspiration for these researchers is human information processing and behavior. Earlier AI systems were rule-based; their decisions were predetermined and designed by the programmer based on inputs. For instance, a rule-based system for medical diagnoses might use a set of rules to match symptoms with specific diseases. Nevertheless, as AI research progressed, it became evident that rule-based systems proved inadequate for capturing the complexity and adaptability of human intelligence. Consequently, researchers embarked on exploring novel AI approaches, such as neural networks and deep learning, to tackle these challenges. The change in AI research directions shows a continuous change in our general understanding of intelligence.

Before characterizing artificial and human intelligence, it is important to first review their definitions.

1.1.1 Definitions of Intelligence

Today, the subject of intelligence is still in debate. Legg and Hutter [2007a] parsed the literature for several definitions of intelligence. Several of these definitions, especially those given by AI researchers, describe intelligence as a property of an agent performing tasks within an environment. The consensus among these definitions is that capacity is the capacity to perform tasks or achieve goals within

an environment. Some definitions involve learning efficiency and skill acquisition as criteria, while others involve adaptation to changes. Certain human cognitive abilities, such as understanding, reasoning, memorization, and imagination, were mentioned as factors of intelligence. Interestingly, abstract thinking is also included among these cognitive abilities.

Legg and Hutter [2007b] also proposes a definition and a formalization of machine intelligence: "Intelligence measures an agent's ability to achieve goals in a wide range of environments." An important aspect of this definition is the variety of goals and environments, which stresses the agent's robustness and capacity to transfer knowledge across environments. A more recent definition by Chollet [2019] describes intelligence as "a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty." This latter definition extends the former with efficiency over skill acquisition and contextualizes the frame of reference with priors on the agent and the environment.

While many definitions of intelligence can be valid, they are useful insofar as they are actionable and essentially descriptive of the skills and evaluation methods associated with intelligence. A definition that focuses on the capacity of a system to perform tasks is not incorrect, but it does not differentiate systems based on their efficiency at solving tasks. On these accounts, Chollet [2019] complements the definition with a formal framework for evaluating systems and a scope of reference for human intelligence with descriptions of human priors. Intelligence is not limited to learning skills; learning and inference efficiency are crucial aspects of intelligence. In the context of human intelligence, efficiency over computational resources, time, and energy distinguishes two individuals with similar capacities for learning skills. Furthermore, the difficulty of learning and performing a task depends on the system's priors, past curriculum, and the task's generalization difficulty. A fair evaluation of different systems should account for these factors.

While most definitions provide a correct depiction of intelligence, they describe intelligence by its consequence; the definition is based on what an intelligent system is capable of doing and not on the process that produces the result. Taking the latter into account as a basis brings a different perspective on intelligence. A general conclusion from most definitions is that intelligence characterizes the mental processes behind actions, performing tasks, and acquiring skills, not merely their execution. The mental processes can be considered information processing operations. From this point of view, intelligence can be characterized

by "*the efficient and productive organization and manipulation of information*".

This new definition implies that the system's intelligence is only tied to its capacity to produce accurate, efficient, and creative solutions to the proposed tasks. The view of intelligence as efficient information manipulation can be reached under assumptions of the system's constraints within its environment, the system's priors, such as its computational resources, interactions within the environment, and the environment's structure.

As a system interacts with its environment, it receives information in forms that it is predisposed to process, following its priors, and selects actions. Depending on the environment's structure, the system's intelligence can be measured by its success at performing tasks since its inception and throughout its lifetime within the environment. A perfect system solves all tasks throughout its existence within the environment and does not need learning. In a practical scenario, a system cannot solve tasks at initialization. Its capacity to solve tasks hinges on understanding tasks and acquiring skills from experience, which highly depend on its information processing proficiency. Given the system's limited computational resources, the system's efficiency at manipulating information determines its level of intelligence. Following this definition, learning can be considered a form of efficiency as it reduces the time and resources used for solving a task. Creativity, which amounts to the meaningful combination of prior knowledge for generating information, is also an important aspect of intelligence as it can be a tool for finding efficient solutions to novel problems. Abstraction can be considered an example of the system's creativity for generalization. The system creates novel concept representations to generalize other related concepts. Overall, the intelligence of a system is not measured by its final performance on a task alone; it also depends on:

- Learning efficiency: the amount of experience needed by the system to reach its maximum performance on a task.
- Time efficiency: the average time spent solving a task instance.
- Resource efficiency: the computational resources used for solving the task.
- Energy efficiency: the amount of energy consumed while solving the task.

By analyzing these aspects in the context of a given environment and system, we can deduce the characteristics of intelligent behavior.

1.1.2 Characteristics of Human and Machine Intelligence

Many aspects of our environment determine features of intelligent behavior. Humans live in a highly complex and dynamic, open-ended environment with only partial access to information. Importantly, a general characteristic of our environment is its compositional nature. Elements of the environment, i.e., objects with which we interact, are built hierarchically as compositions of their components. Within this environment, humans also have limits and constraints in terms of energy, time, and computational resources.

Human intelligence in this context depends on physical predispositions for parsing information, senses, and priors built into the structure of the brain throughout evolution, and the capacity to learn from experience. Faced with highly complex, low-level information, The first characteristic of human intelligence is *abstraction*: the ability to organize information by filtering task-irrelevant variables and recognizing patterns extracted from experience. The partial access to information forces the emergence of many strategies for inferring task-relevant information: probing the environment, using more learning experience, or inferring unknowns based on available information and past experiences. These strategies are traded off to improve overall efficiency. Importantly, humans infer unknown information by building a task-specific model of the environment. This is an important characteristic of human intelligence since *model building* is used for many purposes, including planning actions, using simulation to build hypotheses, and generating experience for learning. Given the high complexity of the environment, inferences over certain tasks can strain the limited resources of the system. Humans trade off accuracy on these tasks with efficiency over the use of their computational resources by developing approximate solutions to inference problems. *Fast approximate inference* is also an important characteristic of intelligence in a system with limited resources.

These characteristics of intelligence are demonstrated in various phenomena of human behavior, such as multi-modal reasoning, causal inference, meta-learning (learning an efficient method for learning new tasks), learning by imitation, zero-shot inference, selective exploration, and trading off on exploration and exploitation. Similar ideas on human intelligence are also shared by prior work [Lake et al., 2016], which focuses on priors, world model building, and fast inference. Griffiths [2020] discusses these ideas from a different angle; he characterizes human intelligence by its limitations. These limitations in time and computation

promote the development of efficient learning and inference mechanisms. Another important aspect of intelligence that we have not discussed thus far is communication. Griffiths [2020] considers communication a limitation since humans are limited in their capacity to transfer information to others, which promotes the development of mechanisms that support cumulative cultural evolution.

An important factor that influences human proficiency in these general skills, and consequently intelligence, is the learning curriculum. Humans can improve their performance on a task with more training, but a training curriculum with increasing difficulty and complexity over trials improves learning speed. Humans can also learn a task without instructions through trial and error. However, being skilled at certain tasks makes completing a task that composes them easier. The diversity of tasks in a curriculum also influences learning speed; a task-diverse curriculum improves generalization and learning efficiency on future tasks. For example, humans are capable of solving certain tasks given only instructions on how to solve them because they have learned how to compose skills for solving new tasks. Many studies support these ideas; they show that formal education, which is a form of curriculum, improves human learning and their generalization of logical reasoning [Attridge et al., 2016, Inglis and Simpson, 2004, Cresswell and Speelman, 2020, Nam and McClelland, 2023].

Many of these concepts have influenced sub-fields of AI research. The sub-fields of meta-learning, multi-task learning, transfer learning, and few-shot learning explore the model capacities for learning many tasks through a sample efficient general learning procedure and by transferring skills across tasks. In continual learning, models are evaluated for their capacity to learn tasks sequentially without catastrophic forgetting. In deep learning, models are compared primarily based on their performance. Some studies in these fields compare the models based on 1) the computational resources measured by model size, 2) the computation time measured by inference time, and 3) energy consumption measured by FLOPs. However, they are mostly secondary factors of comparison. Other sub-fields focus on the optimization of these aspects in neural networks, such as the architecture search, pruning, and quantization literature that explore energy and computation-efficient models. Various techniques for improving or facilitating learning in neural networks have spawned other sub-fields to explore, such as imitation learning, world models, and curriculum learning.

Despite the tremendous progress in these various sub-fields, neural networks remain limited; they are regarded as good models for the fast inference capacities

of humans at best. For example, deep convolutional architectures [Krizhevsky et al., 2017, He et al., 2015] model the object recognition of the ventral visual stream in the brain. This process is hypothesized to involve primarily feedforward propagation of visual information to extract object categories [Eberhardt et al., 2016, Yamins et al., 2014, Rajalingham et al., 2015]. Neural networks suffer from slow and data-inefficient training, a lack of robustness to out-of-distribution settings [Geirhos et al., 2020a], biases towards statistical trends in the data, catastrophic forgetting, and a lack of compositional generalization. These limitations could be caused by many factors, including the lack of flexibility in ANN computations and the disparity in learning strategies and experiences between brains and ANNs. Foundation models such as large language models (LLMs) [Brown et al., 2020, Touvron et al., 2023a,b] and large multi-modal architectures [OpenAI, 2023, Driess et al., 2023] address many of these problems through scale at exorbitant computation and energy costs. However, their capacities remain limited compared to humans [Kaddour et al., 2023] since they display sub-human performance on many logical reasoning tasks and have unreliable outputs since they vary highly depending on their inputs.

A popular hypothesis in the field attributes the lack of reliability, generalization, and flexibility to their failure to implement compositional computation. Due to the compositional nature of our environment, humans are believed to leverage compositionality as a basis for representation and computation. These ideas are shared by Lake et al. [2016] which lists compositionality as an important feature of human intelligence. Smolensky et al. [2022] attributes human intelligence to continuity and compositionality in neural computing, while the absence of continuity explains the failure of earlier symbolic AI systems, the absence or lack of compositionality explains the failure of modern neural network-based systems.

This PhD dissertation explores compositionality as a central aspect of human intelligence. In the following, I delve into its definition and its hypothetical role in supporting other aspects of intelligence. Later, I review works that investigate compositionality in ANNs and integrate them into neural computation. The ultimate goal of this research is to propose a novel neural architecture that follows the principles of human intelligence, exploring how to build and train such a model to leverage the power of compositionality and potentially enhance the capabilities of AI systems to perform complex tasks with greater efficiency and adaptability.

1.2 Compositionality

Abstraction and compositionality are foundational concepts in linguistics and cognitive science that play vital roles in understanding how humans generate and interpret complex linguistic expressions and organize their knowledge. These concepts have evolved, acquiring nuanced definitions and formalizations that have led to rich interdisciplinary discussions and their integration with other key concepts in the cognitive science literature. Compositionality emerged as a crucial concept in formal semantics and the philosophy of language. The principle of compositionality states that the meaning of a complex expression is a function of the meaning of its parts and how they are combined [Frege, 1980]. Here, it is viewed as a property of language and meaning, and it entails that any expression in a compositional language can be understood only using the meanings of its parts and their syntactic structure. Since its introduction, it has been the basis for a large body of work and a topic of contention in linguistics and cognitive science. Various definitions were attributed to compositionality to either broaden its scope with respect to language and meaning or to account for specific cases that the classical definition of compositionality fails to explain. While Partee [1984] provides a global definition of compositionality that does not put constraints on the meanings of parts, the structure and function that combines them, in local compositionality, the meaning of the expression depends only on the largest parts [Szabó, 2012]. Szabó [2022] reviews several definitions with their formalizations and discusses arguments for and against compositionality.

Proponents of this theory motivate it with the notions of productivity, the capacity to produce and understand new expressions in a language given the meanings of the parts of new sentences and the syntax of the language, and systematicity, the ability to generalize syntax across expressions, for example, "the cup is on the table" and "the pillow is on the bed." Productivity and systematicity are strong arguments, but compositionality still fails to explain other phenomena in natural language, such as context-dependence; an expression's meaning might depend on earlier expressions, the overall topic, the priors, and the intentions of the writer and the reader. Idioms are another example of a notion that compositionality does not explain; for example, the meaning of the expression "break a leg" can have different meanings based on context, and its meaning as "good luck" is not inferred from the meanings of its parts.

Beyond debates on compositionality as a property of natural language, it has

also been regarded as a property of human language competence [Pinker, 1984, Fodor and Pylyshyn, 1988, Baroni, 2019, Marcus, 2003]. These views focus on the mind's capacity to process nested structures as the basis for the rich expressiveness and open-ended nature of language [Chomsky, 1957, Dehaene et al., 2015, Hauser et al., 2002].

In this line of work, the "language of thought" theory [Fodor, 1975] claims that thoughts are expressions of a mental language. Several proponents of this theory stipulate that LOT is compositional. The reasoning concludes that the compositionality of LOT is derived from the compositionality of natural language. The same arguments about systematicity and productivity of thought are used to support this theory. The difference between the two, however, is the vagueness of constituent parts in a mental expression. Rescorla [2019] reviews formalizations of expressions in this language. Although theorists present compelling evidence, demonstrations of such ideas in biological systems remain a significant challenge, especially given that LOT modeled mental activity as rule-governed symbol manipulation, which is difficult to translate into biological hardware. In the context of the LOT hypothesis, Frankland and Greene [2020] investigates the neuroscience literature for mechanisms of compositional computation in the brain; however, no clear commitment to a formal framework of compositionality was specified in the paper. Similarly, in an attempt to bridge the gap between psycholinguistic and cognitive science perspectives on compositionality, Baggio [2021] proposes an architecture of the language processing system based on studies from neuroscience. In this context, the principle of compositionality is reframed as a constraint on the architecture and a description of the system's semantic competence.

In the early decades of AI research, compositionality was an explicit feature of AI models since it was the basis for symbolic architecture. Today, more liberal uses of the term can be found in the field. Compositionality can be a feature of the data and representations, a property of the model [Chang et al., 2019, Ringstrom, 2022] or a nature of computation [Kurth-Nelson et al., 2023, Lake et al., 2015, Ellis et al., 2020]. Even though these notions eventually refer to the same idea of global compositionality, their applications remain more general. For example, the representations concern inputs of various types, including images, 3D shapes, tasks, and programs, among others. Among studies that address compositionality as a function or a nature of computation [Schwartenbeck et al., 2021, Kurth-Nelson et al., 2023, McNamee et al., 2022] explain brain function using computational models of the brain function. Other studies on language mod-

els [Chaabouni et al., 2020, Kharitonov and Baroni, 2020] use other operational definitions of compositionality to study language emergence. Caucheteux et al. [2021], Caucheteux and King [2022] follow a different definition of compositionality as a property of the representation that allows for comparing humans and machines in language tasks.

Given this multidisciplinary interest in compositionality, it is natural that definitions would vary depending on the application. A single definition is unlikely to be suitable for all contexts. Nevertheless, being an intrinsic property of the world, compositionality is a general concept that could describe any physical or abstract entity, and its definition should be generalizable. For a biological system to interact with the world, it needs to reason over the compositional nature of the world and learn from experience within its environment. We believe that for the purpose of understanding a highly complex and stochastic biological system such as the brain and its interaction with the world, the theoretical framework of study should be general and constrain the system only with respect to its capacity. Thus, in the following, I will give a high-level definition of abstraction and compositionality.

In general, **abstraction** can be understood as the capacity to conceptualize experience in a mental representation. The brain can build many representations of the number one: a visual representation of "1", different language representations in many languages, a representation of its vocal pronunciation, a representation of the motor sequences that allow for writing the number, and other representations for its use in numerosity and mathematics. This representation allows the brain to identify new instances of the same concept. Importantly, the brain also links these representations with an abstract representation that is dissociated from the concept's multi-modal features. The brain's capacity to create and manipulate these abstract and feature-relevant representations is important for their generalization to new contexts of their use. This capacity is also the basis for compositionality.

Compositionality can be regarded as a method for representing and manipulating concepts within a structure that describes relationships between them. It complements abstraction by generalizing relationships between concepts. The structure is defined in this context as a set of roles and relations between them that can be used for inferences over many concepts. For example, the abstract representation of number one and its multi-model representations are contents placed within the links that constitute a structure. This structure can be used for representing other numbers, such as two and three. The structure-content formulation can be used in the classical meaning of compositionality to represent the com-

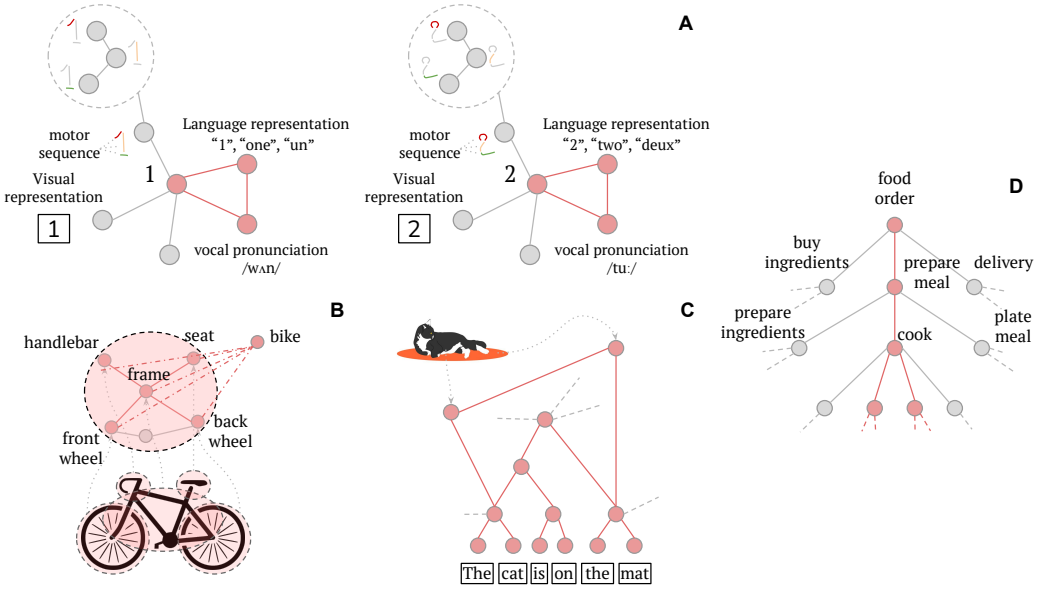


Figure 1.1: **Structure-content separation**: compositional computation consists of a separation between the representation of an abstract concept from its multimodal representations and a separation of these representations from the relationships between them.

position of a concept as a combination of its elementary components within a structure. For example, the motor sequence for writing 1 can be decomposed into the elements that generate strokes in a structure that specifies their sequence and spatial locations.

This formulation of compositionality can be generalized to arbitrary concepts and structures. A system that uses compositional computation can extract abstractions over its internal states, which allows for creating abstractions over structures and structure-content associations. Grounding highly abstract concepts in concrete, low-level concepts facilitates instantiating complex and highly abstract concepts for the system. For example, the set of natural numbers is an abstract concept built on top of numbers such as 1 and the addition operation; grounding its representation in the symbol $\mathbb{N}$ allows for its efficient instantiation.

1.2.1 The Role of Compositionality in Intelligent Systems

To explain the importance given to this notion in various disciplines, we discuss the advantages of using compositionality in a computational system to model the brain. Any system needs to understand the structure of its environment to reach its goals. Given the inherent compositionality of the world, it would be useful as a principle for organizing information; this includes perceptual inputs, actions, tasks, and even internal states of the system. Decomposing information gives one the capacity to analyze components and gain an understanding of the individual factors of variation that underlie them. Taking vision as an example, an interpretation of visual input must take into account the factors of variation that underlie it. This task would be difficult without decomposing a scene into individual components, including objects and sources of light, while considering the perspective of the viewer and inferring the factors of variation of each component, such as positions, colors, and textures. Lee and Mumford [2003], Yuille and Kersten [2006], Parr et al. [2021] support this view and propose a model of visual perception in the brain based on probabilistic graphical models. This understanding of perception can be extended to other modalities, even to action and planning.

Among the arguments supporting the compositionality of natural language is *productivity*. Productivity describes the capacity to generate an infinite number of concepts from a finite set of atomic units. In LOT theory, this translates to the infinite thoughts a human brain can generate while being limited in the number of neurons used to generate them. This is a significant advantage of compositionality because it underlies creativity and expressive power without sacrificing efficiency. In our framework, expressivity is driven by the process of creating abstractions over structures and generalizations through inference over existing structures. A second advantage of compositional computation is the summarized and interpretable organization of information. It facilitates the usage and storage of information for the system and aids in the development of artificial systems by giving users an understanding of the inner processes of the system. Most importantly, systems that leverage compositionality are more efficient in various ways. 1. Learning efficiency: by decomposing tasks into simple sub-tasks, the system can learn and update its knowledge incrementally and reuse the knowledge for future tasks. Furthermore, decomposing a task allows the system to identify errors more easily during execution and channel the rectification or learning signal correctly. 2. Efficiency in learning experience: generalization and flexibility can also be considered as efficiency in the amount of experience needed for learning.

If a system generalizes a function across contexts, it does not need to relearn the function for each context. In the case of compositional generalization, the system can even infer the structure of a novel task and solve it without experience. 3. Efficiency in computational resources through modularity: systems that modularize functions during learning and compose them during inference do not waste computational resources in learning complex functions separately when they share computations.

These theoretical views find applications in several behaviorally relevant functions. **Building world models** To understand its environment, the system could build a model of its components and their interactions. This view of the mind dates back to 1948 [Tolman, 1948], where Tolman theorized that humans and animals form an approximate 'cognitive map' of their environment. This idea is still popular today [Friston et al., 2021] and has applications in neuroscience and machine learning, for example in physical scene understanding [Battaglia et al., 2013] and model-based RL [Ha and Schmidhuber, 2018]. Creating a model of the environment allows the system to simulate actions and outcomes, form plans at various timescales and levels of abstraction, and then select the plans with the highest likelihood of success in the task. Compositionality would not be useful only as a basis for representing a model of the environment but also the states in different simulations, the goals, the predicted rewards, and the plan of action. Furthermore, it would be useful for correcting the model using experience since the model is built on structured representations that can be manipulated by the system.

Reasoning and handling uncertainty When the system has limited access to information within the environment, its available information can be insufficient to solve the task. Compositional representations of the environment in the world model allow the system to know which variables are unknown. Within the world model, it can reason over unknowns with evidence accumulated from the environment and judge its confidence in hypotheses that it has inferred. The level of confidence in the hypotheses can influence the system's decision-making.

Robustness Compositional representations and abstraction can improve the robustness of a system. Here, we consider robustness to be the capacity of the system to maintain performance on a task in uncommon contexts or under variations of the input that do not impact the task structure. If we take visual perception as an example, a system that leverages compositionality could decompose visual scenes based on the structure described above. The compositional representa-

tion and abstraction allow the system to separate factors of variation in individual components of visual input. This concept is known in machine learning as disentanglement [Bengio et al., 2014]. When the system encounters an uncommon situation (such as novel illumination colors or backgrounds), the interpretation of the scene in a compositional representation allows the system to recognize the novelty and perform tasks correctly if they are irrelevant to the novelty (recognizing an object placed in an uncommon background). In machine learning, this is known as out-of-distribution generalization [Shen et al., 2021]. Even when the system makes errors due to incorrect task structure inference, for example, the compositional representation allows for reflecting on the execution, identifying the error, and correcting it in a sample-efficient manner.

Creativity Compositionality provides mechanisms for creating new concepts by combining other known concepts in a structured representation. The benefits of these mechanisms can be observed in problem-solving, for example. Random exploration of strategies for solving a problem is inefficient, but guiding the random exploration into structured representations with priors results in a more efficient search.

Efficient learning and generalization Building world models is important not only for inference but also for learning. Learning new tasks from trial and error or examples could require large amounts of trials, depending on task complexity. The system's proficiency to meta-learn can significantly reduce the amount of experience required for learning new tasks. This consists of inferring task structure, decomposing the task into sub-tasks using prior knowledge, and testing hypotheses. A compositional representation of the task and the functions used for solving it allows the system to identify components that require adjustment. By saving successful plans and procedures, the system can adjust the components that it has identified for learning. If employed in this manner, compositionality could be the key to solving the *catastrophic forgetting* problem encountered in continual learning and the problem of *credit assignment* connectionist systems. We believe that meta-learning on a system that uses compositional computation can significantly improve the learning efficiency Lake et al. [2016].

In this section, I did not make assumptions about the computational tools used by the system to implement notions such as meta-learning, probabilistic inference, or compositional representations. I also believe that a system does not need to rely on compositionality solely as a computational principle.

1.2.2 Compositionality in AI

Compositionality has been a central concept in AI research for decades. In the debates on connectionism and symbolic structure of cognitive architecture, neural networks, as connectionist systems, were criticized for lacking compositional symbol manipulation [Fodor and Pylyshyn, 1988, Lake et al., 2016, Lake and Baroni, 2018, Marcus, 2018]. Many have tested the ability of neural networks to solve tasks requiring compositional generalization, with mixed results [Christiansen and Chater, 1994, Marcus, 1998, Botvinick and Plaut, 2006, Bowers et al., 2009, Botvinick and Plaut, 2009, Frank et al., 2009, Bowman et al., 2015, Frank, 2014]. There were also attempts at developing a schema for representing compositional structures using vectors [Smolensky, 1990]. Ideas about compositionality were adopted to explain modalities beyond language and thought. For example, Hoffman and Richards [1984], Biederman [1985] theorized that the visual system decomposes objects into their parts. In recent years, the field has largely expanded in many directions, developing formalizations and benchmarks for evaluating compositionality in ANNs, probing models for compositional structure, and improving compositional generalization with novel architectures or special training schemes.

The first idea to disambiguate in this research question is the nature of compositionality in ANNs. Early research on visual tasks, namely visual classification, has brought convolutional deep architectures to the forefront. CNNs have since become the standard vision models, and their success has been attributed in part to their capacity to extract hierarchical features from images. This hierarchy of features has been deemed a feature of compositionality [Zeiler and Fergus, 2013]. Although CNNs have a structure that represents features at different levels of abstraction, this structure remains restricted to the representation of image features. For example, standard CNNs cannot decompose a scene into objects and their components. Thus, even if a CNN's structure is considered compositional, its representations are restricted in their usefulness in tasks that involve compositionality. The compositional structure can also be built into the data and the process, as in recursive neural networks [Socher et al., 2013]. These examples highlight important questions: what does compositionality characterize in neural networks, the representations, or the structure of the model? Is it implicit or explicit? And to what extent is it generalizable?

Given the difficulty of organizing diverse data into a compositional format, the

field has focused on investigating the implicit compositionality in the representations and weights of neural architectures and building compositionality into the internal structure of the model. Compositional generalization has been investigated in various contexts: zero-shot learning in vision [Yang et al., 2020, Mancini et al., 2021, Misra et al., 2017, Naeem et al., 2021, Purushwalkam et al., 2019, Atzmon et al., 2020, Wang et al., 2020], 3D representations [Tulsiani et al., 2018], visual reasoning [Johnson et al., 2017a], reinforcement learning [Gur et al., 2022], language [Lake and Baroni, 2018, Keysers et al., 2020] and abstract tasks such as math [Saxton et al., 2019]. In most settings, models are evaluated based on systematicity, whereby novel combinations of features are introduced during testing. Results in these studies vary, with a trend towards the failure of standard models at compositional generalization.

We take the SCAN dataset as an example. SCAN tasks involve translating commands from a simplified natural language to a sequence of actions. Results show that standard recurrent models do not learn these tasks compositionally [Loula et al., 2018, Lake and Baroni, 2018]. Although pre-training masked language models have better performance [Furrer et al., 2021], they still do not learn compositionally. However, Lake and Piantadosi [2019] shows that augmenting a seq2seq architecture with memory allows it to solve many SCAN tests.

Given that most compositional generalization tests are limited to one aspect of compositionality, which is systematicity, Hupkes et al. [2020] proposes PCFG SET, a suite of tests for five aspects of compositionality: systematicity, productivity, locality vs. globality, substitutivity, and overgeneralization. Their analysis of standard architectures shows that they fail at most tests.

Improvements to the compositionality of neural network models vary between training schemes and the use of different inductive biases. Baan et al. [2019], Hupkes et al. [2019] show that training models with attentive guidance biases them to implement more compositional solutions and improve compositional generalization. Attentive guidance is implemented as an additional supervision signal that expresses how the input should be segmented and in which order it should be processed. In a reinforcement learning task, Hill et al. [2020] shows that increasing the perceptual variety and realism of the environment improves compositional language generalization. These examples show that training experience influences compositional behavior in neural networks and can even bias models to implement compositional computations.

Modular architectures have been used to implement explicitly compositional computation in many scenarios [Andreas et al., 2016a, Hu et al., 2017]. Some approaches use program induction with program primitives [Johnson et al., 2017a]. These approaches require the implementation of a diverse set of program primitives and are limited in their capacity to learn new programs. Inspirations from brain function were used for implementing other approaches [Russin et al., 2019], but they have limited improvements over standard architectures.

Modern large-scale deep learning architectures, such as generative models [Ramesh et al., 2022, Rombach et al., 2022] Large language models [Brown et al., 2020, Touvron et al., 2023a,b] and multi-modal foundation models [OpenAI, 2023, Driess et al., 2023, Yu et al., 2022], show impressive capacities in various tasks in zero-shot settings. They seemingly present scale as the correct solution for generalization. However, beyond exorbitant training and inference costs, their failures demonstrate brittleness and sub-human performance on many reasoning tasks. Their limitations on compositionality have been demonstrated in various scenarios [Dziri et al., 2023].

This brief overview of research on compositionality highlights that, to date, there are no deep learning models that perform well reliably in compositional generalization tests. Current architectures rely on unique inductive biases that limit their expressivity and their ability to represent diverse compositional structures. Furthermore, compositionality benchmarks explore predominantly tests of compositionality during inference. To our knowledge, models have not been evaluated for their capacity to decompose tasks during learning.

1.3 Original Contributions

The goal of this dissertation is to promote a direction of research that focuses on all aspects of human intelligence, especially compositionality. My work attempts to address the shortcomings of deep learning models: their reliance on large data, their lack of generalization, and their robustness. In the first chapter, I attribute these shortcomings to the lack of focus on factors of human intelligence in the evaluation of deep learning models: efficiency in time, computational resources, energy, and learning experience. Investigating factors of intelligence, I focus on compositionality and specifically compositional learning—the system's capacity to decompose tasks into their elementary components during training and compos-

ing learned skills to learn new tasks efficiently. Compositionality in deep learning is often regarded as a mechanism that models leverage during inference and not during learning. These ideas motivate developing a benchmark for evaluating compositional learning and sample efficiency.

Chapter 2 details the development of CVR, the "compositional visual reasoning" benchmark, which includes 103 tasks built as compositions of nine elementary visual relations. To build this benchmark, I propose a novel method for creating visual reasoning problems with a compositional prior.

Chapter 3 focuses on the evaluation of several baseline models for sample efficiency, compositional learning, and out-of-distribution generalization. The results demonstrate a gap between humans and models in terms of sample efficiency, the failure of models to generalize to out-of-distribution settings, and their failure to decompose tasks into their elementary components during training.

This dissertation recognizes the generality of compositionality as a computational paradigm, its multipurpose use by the brain in a variety of functions, and its nature as an emergent property of brain function. Given these observations, I reason that compositionality could emerge in a neural architecture that takes inspiration from brain functions. Furthermore, an architecture that captures the flexibility of brain functions could benefit from the use of compositionality in inference and learning for representing abstract concepts and factorizing computation, thus stepping closer to human intelligence.

In Chapter 4, I propose a framework for developing and training neural architecture based on principles taken from the brain's biological priors and emergent properties. These principles include modularity, agency, control over internal computations, learning rules, credit assignment, curricular organization, and diversity of tasks in the learning experience. Using some of these principles, I propose AbstractNet, a modular architecture with control over internal computations, information routing, module activations, and adaptive computation time. Initial experiments show that AbstractNet can learn how to use diverse module architectures to solve a variety of homogeneous and heterogeneous tasks and adapt its internal computations based on task demands. The implementation ideas proposed in the framework could potentially further improve this architecture and expand its capabilities.

Overall, this dissertation shows that compositionality is a missing component in deep learning models for reaching human-level intelligence and proposes a

framework for incorporating brain-inspired design principles in neural architectures.

The work presented in Chapter 2 and Chapter 3 is adapted and expanded from the following publication:

- **Aimen Zerroug**, Mohit Vaishnav, Julien Colin, Sebastian Musslick, and Thomas Serre. "A benchmark for compositional visual reasoning." *Advances in Neural Information Processing Systems 35* 2022 29776-29788.

CHAPTER 2

BENCHMARKING COMPOSITIONALITY AND SAMPLE EFFICIENCY

2.1 Introduction

Visual reasoning is a complex ability requiring a high level of abstraction over high-dimensional sensory input. It highlights humans' capacity to manipulate concepts and relations as symbols extracted from visual input. The efficiency with which humans learn new visual concepts and relations, as exemplified by fluid intelligence and non-verbal reasoning tests, is equally fascinating. In the pursuit of human-level artificial intelligence, a growing body of research is attempting to emulate this skill in machines, and deep neural networks are at the forefront of the field.

Deep learning approaches are prime candidates as models of human intelligence due to their success at learning from data while relying on simple design principles. However, these architectures are imperfect models of human intelligence, as shown by their lack of sample efficiency, inability to generalize to unfamiliar situations [Geirhos et al., 2020b], and lack of robustness [Goodfellow et al., 2014]. Their ability to perform well in large-data regimes has skewed researchers to scale up datasets and architectures with little consideration for the sample efficiency of these systems.

Only a few benchmarks address these aspects of human intelligence. One such benchmark, ARC [Chollet, 2019], provides diverse visual reasoning problems. However, the extreme scarcity of training samples—only 3 samples per task—makes the benchmark difficult for all methods, especially neural networks. Other benchmarks have led to the development of new neural network-based models that address particular gaps between human and machine intelligence [Barrett et al., 2018, Zhang et al., 2019, Fleuret et al., 2011]. Some focus on evaluating the task's perceptual requirements [Fleuret et al., 2011], which include detecting features, recognizing objects, perceptual grouping, and spatial reasoning. Others evaluate logical reasoning requirements [Barrett et al., 2018, Zhang et al., 2019], such as symbolic reasoning, making analogies, and causal reasoning. However, they lack either the variety of abstract relations present in the scene or the semantic and structural variety of scenes over which they instantiate these abstract relations.

Creating novel visual reasoning tasks can be challenging. In this benchmark, we standardize a process for creating tasks compositionally based on an elementary set of relations and abstractions. This process allows us to exploit a wide

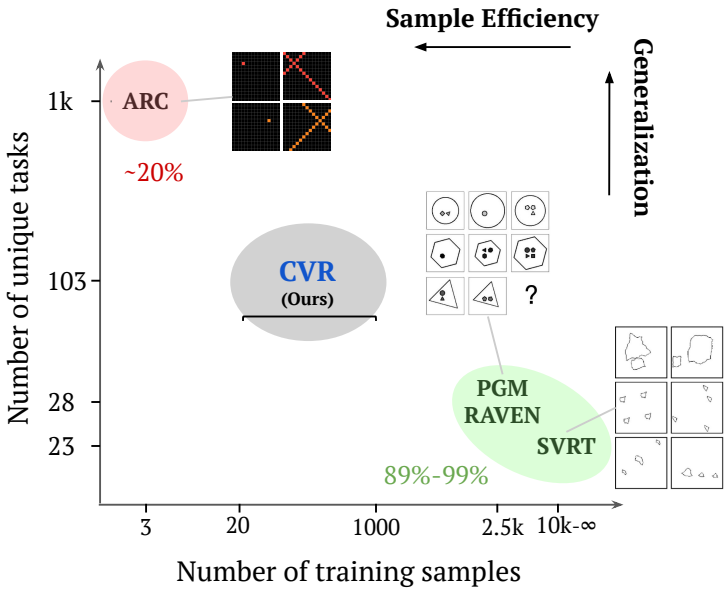


Figure 2.1: **Visual reasoning benchmarks**: State-of-the-art models achieve super-human accuracy [Wu et al., 2020, Vaishnav et al., 2022] on several visual-reasoning benchmarks such as RAVEN [Zhang et al., 2019], PGM [Barrett et al., 2018], and SVRT [Fleuret et al., 2011]. However, some benchmarks continue to pose a challenge for current models, such as ARC [Chollet, 2019]. The fundamental difference between these different benchmarks is the number of unique task rules they composed out of their priors and the number of samples available for training architectures on individual rules. This difference sheds light on two poorly researched aspects of human intelligence: learning in low-sample regimes and harnessing compositionality. The proposed CVR challenge aims to fill the gap between current benchmarks to encourage the development of more sample-efficient and more versatile neural architectures for visual reasoning.

range of visual relations as well as abstract rules, thus making it possible to evaluate both the perceptual and logical requirements of visual reasoning. The compositional nature of the tasks provides an opportunity to investigate the learning strategies wielded by existing methods.

Contributions Our contributions can be summarized as follows:

- A novel visual reasoning benchmark called **Compositional Visual Rela-**

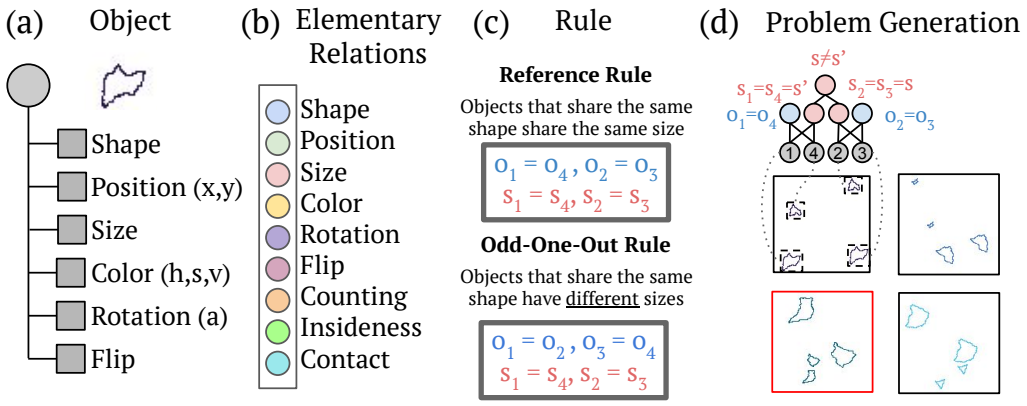


Figure 2.2: **Scene Generation**: A scene in our image dataset is composed of objects. (a) An object is a closed contour with several attributes. (b) A relation is a constraint for the generation process over scene attributes. (c) The elementary relations control unique scene attributes. They are used for building task rules in a compositional manner. Each task uses a reference rule and an odd-one-Out rule to generate images. (d) Odd-one-out problems are randomly generated using a program. Three images are generated following the Reference rule, and a fourth image (highlighted in red) is generated following the Odd-One-Out rule.

tions (CVR) with 103 unique tasks over distinct scene structures.

- A novel method for generating visual reasoning problems with a compositionality prior.

2.2 Visual Reasoning Tasks

CVR is a synthetic visual reasoning dataset that builds on prior AI benchmarks [Fleuret et al., 2011, Chollet, 2019] and is inspired by cognitive science literature [Ullman, 1987] on visual reasoning. In the following, we will describe the generation process of the dataset.

2.2.1 Odd-One-Out

The odd one-out task has been employed in prior work to test visual reasoning [Mańdziuk and Żychowski, 2019]. A sample problem consists of four images generated such that one of them is an outlier according to a rule. The goal of the task is to select the outlier. The learner is expected to test several hypotheses in order to detect the outlier. This process requires them to infer the hidden scene structure and relationships between the objects.

2.2.2 Visual Relations

Each image contains one **scene** composed of multiple **objects** as shown in Figure 2.2. An object is defined as a closed contour with a set of **object attributes**: *shape*, *position*, *size*, *color*, *rotation*, and *flip*. Other attributes describe the scene or low-level relations between objects. *Count* corresponds to the number of objects, groups of objects, or relations. *Insideness* indicates that an object contains another object within its contour. *Contact* indicates that two object contours are touching. These nine attributes are the basis for the nine **elementary relations**. For example, a "size" relation is a constraint on the sizes of certain objects in the scene. Relations are expressed with natural language or logical, relational, and arithmetic operators over scene attributes. Relations and objects are represented as nodes in the **scene graph**. Relations define groups of objects and can have attributes of their own. Thus, it is possible to create abstract relations over these relations' attributes. A scene can be generated from a template that we call a **structure**. The concepts of structure, scene graph, and relations are used to formalize the process behind designing a task. In practice, the **generation process** is a program implemented by the task designer to generate problem samples for one task randomly. The pseudocode for an example program is detailed in Alg. 1.

2.2.3 Rules and Problem Creation

The generation process described above can be used to instantiate different tasks: binary classification, few-shot binary classification, or Raven's progressive matrix. In this paper, we choose to apply this process to create odd-one-out problems. First, the task designer selects target relations and incorporates them into a new

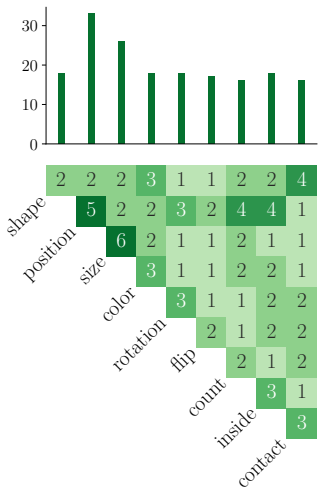


Figure 2.3: **Dataset rules**: Each square represents the number of tasks in CVR that are a composition of one or two elementary relations. Tasks on the diagonal involve complex reasoning over a single elementary relation. The bar plot shows the number of rules that involve each elementary relation.

scene structure. In Figure 2.2, the target relations are size and shape similarity; they are added to a scene with 4 objects. Then, a reference rule and an odd rule are chosen such that they combine target relations in different ways. The reference and odd rules in the example vary only in the size or shape attributes. A valid odd-one-out rule contradicts the reference rule, such that any strategy used to solve the task must involve exclusively reasoning over the target relations. Given a scene structure, a reference, and an odd-one-out rule, the generation process has a set of free parameters that control the generation process for new samples. The problem's difficulty level can be varied by randomizing or fixing these parameters. In the shape-size task, the range of color values and the variation of objects across the 4 images are examples of free parameters. More random parameters result in a higher difficulty. We create generalization test sets by changing the sets of fixed or random parameters. For more details on the generalization test sets, we refer the reader to the annex.

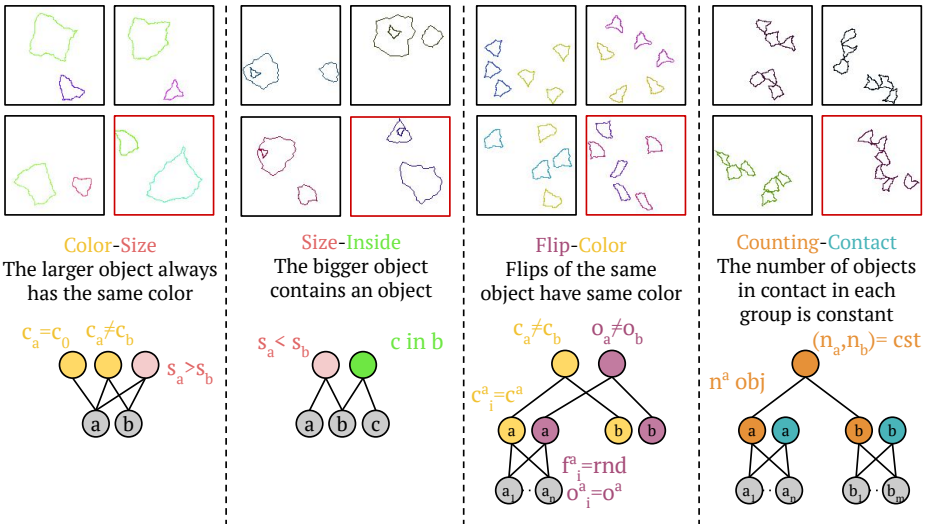


Figure 2.4: **Examples of task rules that are composed of a pair of relations.** More examples of tasks and algorithms are provided in the SI.

2.3 Dataset Details

CVR incorporates 103 unique reference rules, including nine rules instantiating the nine elementary visual relations and 94 additional rules built as compositions. These compositions span all pairs of elementary rules and include up to four relations. While some rules are composed of the same elementary relations, they remain unique in their scene structure or associations with other relations. 20 are compositions of single elementary relations, 65 are compositions of a pair of relations, and 9 are compositions of more than two elementary relations. Figure 2.3 details the number of unique rules for each pair of elementary relations. The procedural generation of problem samples helps us create an arbitrary number of samples. We create 10,000 training problem samples, 500 validation samples, and 1,000 test samples for each task. We also create a generalization test set of 1000 samples.

We define compositionality prior to the task's design constraint, which ensures that solving the task requires reasoning over its elementary components. In the size-shape task, shown in figure 2.2, the outlier can be differentiated from the other images by reasoning purely on size and shape. In the context of CVR, compositionality is not exemplified by combinations of object attributes only, such as

novel color and shape combinations in an object; it is also exemplified by combinations of variables at higher levels of abstraction, such as groups of objects and scene configurations. For example, the position-rotation composition rule in Fig. 2.4 requires reasoning over the rotation properties of two sets of objects in each scene and the position properties of objects within each set.

CVR constitutes a significant extension to the Synthetic Visual Reasoning Test (SVRT) [Fleuret et al., 2011] in that it provides a systematic reorganization based on an explicit compositionality prior. Among the 23 SVRT tasks, many share relations, such as tasks #1 and #21, which both involve shape similarity judgments. Most of these tasks can still be found among CVR's rules. At the same time, CVR is more general because it substitutes binary classification tasks with odd-one-out tasks, which allows for exploring more general versions of these tasks with a broader set of task parameters. For example, in SVRT's task #7, images of 3 groups of 2 same shapes are discriminated from images of 2 groups of 3 same shapes. This task is a special case in CVR of a more general *shape-count* rule with n groups of m objects where the values are randomly sampled across problem samples. Unlike procedurally generated Raven's Progressive Matrice (RPM) benchmarks [Barrett et al., 2018, Zhang et al., 2019], CVR does not rely on a small set of fixed templates for the creation of task rules. The shapes are randomly created, and positions are not fixed on a grid (for most rules), which renders the visual tasks difficult for models that rely on rote memorization [Kim et al., 2018]. Other attributes are sampled uniformly over a continuous interval.

2.3.1 Priors on Scene Generation

- Objects are defined as closed contours sampled randomly with two parameters: the radius of the largest circle that can fit inside the object and the distance between two consecutive points in the closed contour.
- The size of the object scales the width and height of the object linearly.
- The background color is always white, and object colors are sampled in the HSV color space with saturations and values that maintain contour visibility on a white background.
- The center of an object is defined as the midpoint between its horizontal and vertical extents. It is used for placing the object based on its position.

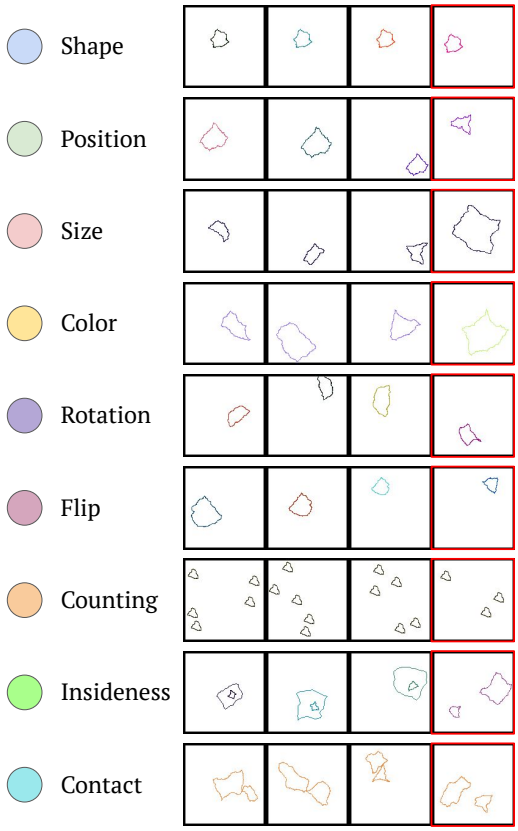


Figure 2.5: **Elementary Tasks** Problem instances from the tasks built based on elementary relations.

> By default, objects are not in contact and do not overlap unless they have an insideness or a contact relationship. To ensure this, their positions are sampled with rejection sampling.

All parameter values are sampled from a fixed range for each task. Within the specified ranges, value combinations might violate priors. Rejection sampling is employed to avoid choosing such value combinations. For example, the object size is sampled such that all objects can fit inside the image based on the number of objects in the scene. The range of sizes decreases with the number of objects in the scene. Functions that sample positions within the scene, positions inside an object, positions of objects in contact, and colors are shared among all 103 programs. Position sampling rejects samples where at least one pair of object-

bounding boxes intersects. To sample contact between two objects, a direction (2D vector) is sampled or specified, then the maximum distance between two intersecting objects is computed, and relative positions with respect to their center of mass are assigned to the objects. While sampling a position inside an object, samples are rejected if the position is outside the object or if contours intersect. When objects contain other objects, their shapes are sampled with a large inner radius. Objects are flipped either horizontally or vertically.

Algorithm 1: Shape-Size Problem Generation: Pseudo-code of the program that generates the shape-size task in Figure 2.2

$n \leftarrow 4$ `// Number of objects`
for $i \leftarrow 1$ **to** 4 **do**
  $s \leftarrow sample\_size()$
  $s' \leftarrow s \times rand([2/3, 1/4])$
  if $i = 4$ **then**
    `// Odd-One-Out`
    $[s_i]^{1-n} \leftarrow [s, s', s, s']$
  else
    $[s_i]^{1-n} \leftarrow [s, s, s', s']$
  end
  $[o, o'] \leftarrow sample\_shapes(n = 2)$
  $[o_i]^{1-n} \leftarrow [o, o, o', o']$
  $[p_i]^{1-n} \leftarrow sample\_position([s_i]^{1-n})$
  $[c_i]^{1-n} \leftarrow sample\_color(n = 1)$
end
$[scene]_{1-4} = [[o, p, s, c]^{1-n}]_{1-4}$
$[image]_{1-4} = [render(scene)]_{1-4}$

Algorithm 2: Size-Color Problem Generation: Pseudo-code of the program that generates the size-color task in Figure 2.4.

$c_0 = sample\_color(n = 1)$
for $i \leftarrow 1$ **to** 4 **do**
  $c_i^a \leftarrow sample\_color(reject = c_0)$
  $c_i^b \leftarrow c_0$
  $s_i^a = sample\_size()$
  $s_i^b = s_i^a \times rand([1/4, 1/2])$
  $[p_i^a, p_i^b] \leftarrow sample\_position([s_i^a, s_i^b])$
  $[o_i^a, o_i^b] = sample\_shapes(n = 2)$
  if $i = 4$ **then**
    `// Odd-One-Out`
    $[c_i^a, c_i^b] \leftarrow [c_i^b, c_i^a]$
  end
end
$[scene]_{1-4} = [[o, p, s, c]^{a-b}]_{1-4}$
$[image]_{1-4} = [render(scene)]_{1-4}$

2.3.2 Generalization Test Set

Among the limitations of our dataset is that certain tasks could be solved by exploiting shortcuts. Shortcuts are biases in the tasks that neural networks exploit to solve them. An explanatory example is the counting task. If objects have the same size, the neural network can easily solve the task by summing the pixels of the image without analyzing the scene. If a model exploits this shortcut, it does not need to learn the concept of an object or counting. To account for this limitation and evaluate out-of-distribution generalization, we develop a generalization test set that differs from the in-distribution test set in several ways.

- **Parameter value ranges**: the ranges used for sampling parameters, especially target-relation parameters, are changed. For example, the range of object numbers for counting related tasks is expanded.
- **Object contour specification**: the distance between dots in contours is randomized locally, resulting in fuzzier contours as shown in Figure 2.6.
- **Sets of random and fixed parameters**: generation parameters that are task irrelevant are generally fixed across the four choice images. In the generalization test set, the sets of random and fixed parameters are changed without affecting the rules. For example, in the shape-size task of Figure 2.2, the color parameter, which is irrelevant, is fixed in the training set and the test set; however, it is randomly sampled in the OOD generalization test set.

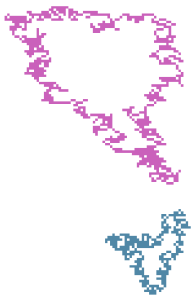


Figure 2.6: **Shapes in the generalization test set.**

2.4 Related Work

2.4.1 Visual reasoning benchmarks

Visual reasoning has been a subject of AI research for decades, and several benchmarks address many relevant tasks. This includes language-guided reasoning benchmarks such as CLEVR [Johnson et al., 2017b], which has been extended in its visual composition by recent work [Li and Søgaard, 2022], physics-based reasoning, and reasoning over time dynamics [Yi et al., 2019, Bakhtin et al., 2019]. Abstract visual reasoning benchmarks are more relevant to our work. Raven's Progressive Matrices (RPMs), which were introduced in 1938 [Burke, 1985], are one example used to test human fluid intelligence. RPMs generally consist of three sequences of three images that describe a logical relationship through specific features such as size, shape, color, and numbers. In the test, the user is prompted to choose the third image of the third sequence among a set of false options. Procedural generation techniques for RPMs [Wang and Su, 2015] enabled the creation of the PGM dataset and RAVEN [Barrett et al., 2018, Zhang et al., 2019]. They also inspired Bongard-Logo [Nie et al., 2020], a concept learning and reasoning benchmark based on Bongard's 100 visual reasoning problems [Bongard, 1968]. Another reasoning dataset, SVRT [Fleuret et al., 2011], focuses on evaluating similarity-based judgment and spatial reasoning. Besides these synthetic datasets, real-world datasets were developed with similar task structures to Bongard-Logo and RPM [Teney et al., 2020, Jiang et al., 2022]. In this work, we take inspiration from SVRT and develop a more extensive set of rules with careful considerations for the choice of rules and using a novel rule generation method. Finally, Abstract Reasoning Corpus [Chollet, 2019] is a general intelligence test introduced with a new methodology for evaluating intelligence and generalization. The numerous problems presented in this benchmark are constructed with a variety of human priors. The unique nature of the task, requiring solvers to generate the answer, and the limited amount of training data render the benchmark difficult for neural network-based methods. We follow a similar approach in our dataset by creating several unique problem templates. However, we restrict the number of samples to a reasonable range to evaluate the sample efficiency of candidate models.

2.4.2 Compositionality

Compositionality is a highly studied topic in AI research. Although there is agreement over the high-level definition of compositionality—the ability to represent new abstractions based on their constituents and their contexts—there is little consensus on methods for characterizing compositional generalization in neural networks. Several tests for compositionality have been proposed in language [Linzen et al., 2016], mathematics [Saxton et al., 2019], logical reasoning and navigation [Bowman et al., 2015, Lake and Baroni, 2018, Ruis et al., 2020, Wu et al., 2021] and visual reasoning Johnson et al. [2017b], Thrush et al. [2022], Agrawal et al. [2017]. Recent work [Hupkes et al., 2020] attempts to identify components of compositionality and proposes a test suit that unifies them. These tests evaluate the model's capacity to manipulate concepts during inference. Systematicity tests the novel combination of features, akin to CLEVR's CoGenT [Johnson et al., 2017b] and C-VQA [Agrawal et al., 2017], where novel combinations of shapes and colors are introduced in the test set, and localism tests the model's ability to account for context similarly to samples from Winoground [Thrush et al., 2022].

2.4.3 Neuroscience and Psychology

Several theories attempt to propose an understanding of the mechanisms behind visual reasoning. Gestalt psychology provides principles hypothesized to be used by the visual system as an initial set of abstractions. Another theory describes visual reasoning as a sequence of elemental operations called visual routines [Ullman, 1987] orchestrated by higher-level cognitive processes. These elemental operations are hypothesized to form the basis for spatial reasoning, same-different judgment, perceptual grouping, contour tracing, and many other visual skills [Cavanagh, 2011]. Evaluating these skills in standard vision models is a recurring subject in machine learning and neuroscience research [Kim et al., 2019, Linsley et al., 2020, Puebla and Bowers, 2021]. To provide a comprehensive evaluation of visual reasoning, it is important to include task sets that require various visual skills within humans' capabilities.

2.5 Discussion and Future Work

In this work, we have proposed a novel benchmark that focuses on two important aspects of human intelligence: compositionality and sample efficiency. Inspired by visual cognition theories [Ullman, 1987], the proposed challenge addresses the limitations of existing benchmarks in the following ways: (1) It extends previous benchmarks by providing a variety of visual reasoning tasks that vary in relations and scene structures; (2) all tasks in the benchmark were designed with compositionality prior, which allows for an in-depth analysis of each model's strengths and weaknesses; and (3) it provides a quantitative measure of sample efficiency.

While CVR is quite extensive in terms of the visual relations it contains, it can always be further improved in its use of elementary visual relations. For example, the shapes could be parametrically generated based on specific geometric features. Hopefully, CVR can be expanded in future work to test more routines by including additional relations borrowed from other, more narrow challenges, including occlusion [Kim et al., 2019], line tracing [Linsley et al., 2018], and physics-based relations. The rules in the current benchmark are limited to 2 or 3 levels of abstraction to evaluate relations systematically. We hope that the release of our benchmark will encourage researchers in the field to test their own model's sample efficiency and compositionality.

CHAPTER 3

EVALUATING COMPOSITIONALITY IN MODERN NEURAL NETWORKS

3.1 Introduction

In this chapter, I focus on the evaluation of deep learning models on the CVR benchmark. This evaluation focuses on state-of-the-art abstract visual reasoning models and standard vision models. These models have been shown to reach high performance on several visual reasoning tasks in previous works [Wu et al., 2020, Vaishnav et al., 2022], but they always require large amounts of data. This paper's subject of interest is quantifying these models' sample efficiency and compositional learning.

This work includes large-scale experiments that capture a multitude of setups, including multi-task and individual task training, pre-training with self-supervision on dataset images to contrast learning of visual representations vs. abstract visual reasoning rules, training over a range of data regimes, testing transfer learning between dataset tasks, and evaluating out-of-distribution generalization. We present an in-depth analysis of task difficulty, which provides insights into the strengths and weaknesses of current models. Overall, we find that the best baselines trained in high-data regimes fall short of human sample efficiency for learning CVR tasks. While models appear to be capable of transferring knowledge across tasks, the results show that they do not leverage compositionality to decompose tasks into their components. We hope to inspire research on more efficient visual reasoning models by releasing our dataset. The code for generating the full dataset and training models is available here.

Contributions Our contributions can be summarized as follows:

- A systematic analysis of the sample efficiency of baseline visual reasoning architectures.
- An empirical study of models' capacity to use compositionality to solve complex problems.
- An evaluation of the out-of-distribution generalization capabilities of baselines.

3.2 Experimental setting

3.2.1 Baseline models

In our experiments, we selected two vision models commonly used in computer vision. We evaluate ResNet [He et al., 2015], a convolutional architecture used as a baseline in several benchmarks [Barrett et al., 2018, Zhang et al., 2019, Vaishnav et al., 2022] and also used as a backbone in standard VQA models. We also evaluate ViT, a transformer-based architecture [Dosovitskiy et al., 2020]. ViT is used for various vision tasks, such as image classification, object recognition, captioning, and recently in visual reasoning on SVRT [Messina et al., 2021]. To compare the architectures fairly, we choose ResNet-50 and ViT-small, which have an equal number of parameters. Additionally, we evaluate two baseline visual reasoning models designed for solving RPMs: SCL [Wu et al., 2020], which boasts state-of-the-art accuracy on RAVEN and PGM, and WReN [Barrett et al., 2018], which is based on a relational reasoning model [Santoro et al., 2017]. Finally, we present SCL-ResNet-18, which consists of an SCL with ResNet as a visual backbone, thus combining ResNet's perception skills with SCL's reasoning skills.

3.2.2 Joint vs. individual rule learning

Models are either trained in a single-task (individual) or multi-task (joint) setting. In the context of the multi-task training on CVR, one image is considered an odd one-out with respect to a reference rule. However, because of the randomness of scene generation, a different image might be considered an odd one-out with respect to a different, irrelevant rule. To illustrate this problem, let's take the elementary size rule as an example. In this rule, each image contains one object. Due to the random sampling of object attributes, it is possible for one image to be considered an outlier with respect to the color rule (the attributes in the 4 images are i-small/green, ii-large/green, iii-small/green, and iv-small/blue). Without specifying that the task to solve involves a size relation, the model could incorrectly choose the fourth image because it is an outlier with respect to the color rule. Thus, models trained on several tasks could easily confound rules. To avoid this problem, models are provided with a rule embedding vector. Given the rule token, models can learn several strategies and use the correct one for each problem

sample. We also compare the multi-task and single-task settings, as they allow for testing the model's capacity and efficiency in learning several strategies and routines to solve different rules. All hyperparameter choices and training details are provided in the Annex.

3.2.3 Self-Supervised pre-training

Unlike humans, who spend a lifetime analyzing visual information, randomly initialized neural networks have no visual experience. To provide a more fair comparison between humans and neural networks, we pre-train baseline models on a subset of the training data. Self-supervised learning (SSL) has seen a rise in popularity due to its usefulness in pre-training models on unlabeled data. By using SSL, we aim to dissociate feature learning from abstract visual reasoning in standard vision models. We pre-trained ViT-small and ResNet-50 on 1 million images from the dataset following MoCo-v3 [Chen et al., 2021a]. In addition to SSL pre-trained models, we also fine-tune models pre-trained on object recognition and image annotation. Since image annotation requires visual reasoning capabilities, these pre-trained models provide a more fair comparison with humans, who regularly perform the task. We select ResNet-50 and ViT-small pre-trained on ImageNet [Deng et al., 2009]. We also pick CLIP [Radford et al., 2021] visual encoders ResNet-50 and ViT-Base, which are trained jointly with a language model on image annotation.

3.2.4 Learning to spot the Odd-One-Out

The training setup for standard vision models is straightforward; models are trained to represent the odd-one-out differently from the three other images. The four images of the problem are fed separately to the model. Their representations are transformed into a low-dimensional space where the distances between the four representations are computed. The cosine similarity of the odd-one-out to the group is minimized with a cross-entropy-based loss. Given the 4 image representations x_i, the logits y_i used for computing the softmax cross-entropy loss are the negative sum of the similarity scores.

$$y_i = -\sum_{j \neq i} \frac{x_i \cdot x_j}{||x_i||_2 \cdot ||x_j||_2}$$

Although we follow a specific training setting for the baseline architectures, we do not impose this methodology on future work as more sophisticated methods for comparing the four images of a problem can be designed.

3.2.5 Human Baseline

As found in Fleuret et al. [2011], having 21 participants solve the 9 tasks based on elementary relations and 36 randomly sampled complex tasks is sufficient to yield a reliable human baseline. We used 20 problem samples for each rule, which corresponds to the lowest number of samples used for training baseline models. We recruited 21 participants from Prolific: 13 females and 8 males aged between 19 and 49 years. All participants signed a consent form before participation and received $10.50 US per hour for participation. The study was approved by the Institutional Review Board of Brown University. 40 individuals were initially enrolled to participate, but 19 were disqualified based on technical malfunctions, misunderstandings of instructions, or failures in attention checks. Participants were instructed to identify the odd stimulus that violated the rule they had to infer over a series of trials. Prior to the practice phase, they were quizzed on their understanding of the task. Participants practiced the task on a separate set of visual stimuli different from the benchmark. During the experiment, participants were informed about the start of each block as well as the concomitant rule switch. For each trial, they were presented with four choices on the screen and instructed to choose the image that seemed to be different according to the rule that they had to learn. They rated their confidence in their choice and received feedback after each trial. In addition, they were asked to describe the rule at the end of each block.

3.3 Results

3.3.1 Sample Efficiency

Baseline models are trained in six data regimes ranging from 20 to 1000 training samples. All sample efficiency results are summarized in Table 3.1. Randomly guessing yields 25% accuracy. We observe that most randomly initialized models are slightly above chance accuracy after training in low data regimes. They

N train samples			20		50		100		200		500		1000	
rand-init	ind	ResNet-50	28.0	1	31.1	1	32.5	3	34.0	6	38.7	12	44.8	24
		ViT-small	28.6	1	30.1	4	30.9	4	31.9	4	33.8	4	35.1	7
		SCL	26.9	0	30.0	1	30.3	2	30.0	2	31.4	2	33.4	5
		WReN	30.0	0	32.0	2	32.9	2	34.1	3	36.3	6	39.0	15
		SCL-ResNet 18	**31.4**	1	**37.3**	9	**37.8**	9	**39.6**	15	**42.7**	21	**48.3**	26
	joint	ResNet-50	**27.5**	0	28.2	0	29.9	2	33.9	6	**52.1**	29	59.2	34
		ViT-small	27.3	1	27.8	2	28.0	1	28.1	1	29.9	2	31.4	3
		SCL	25.8	0	25.8	0	28.3	1	34.1	3	43.2	22	46.2	27
		WReN	26.8	0	27.6	0	28.5	0	30.1	0	36.4	9	42.3	20
		SCL-ResNet 18	26.4	0	**28.4**	0	**31.6**	4	**40.7**	13	51.4	32	**64.0**	42
SSL	ind	ResNet-50	40.5	13	47.3	18	52.9	29	56.8	34	61.9	42	**67.7**	50
		ViT-small	**46.7**	16	**51.6**	24	**54.8**	29	**57.5**	38	**62.0**	44	65.5	46
	joint	ResNet-50	**44.3**	16	**50.3**	24	**55.3**	30	**59.5**	42	**68.9**	49	**79.2**	59
		ViT-small	39.3	15	39.5	13	40.8	14	44.1	16	53.3	30	60.7	41
IN	joint	ResNet-50	**32.0**	2	**35.1**	5	**39.0**	9	**43.8**	13	**57.7**	48	**69.5**	48
		ViT-small	27.9	2	28.2	1	28.6	2	30.0	2	35.6	5	47.2	24
CLIP	joint	ResNet-50	28.7	0	32.0	2	40.8	11	46.9	18	59.7	40	74.4	53
		ViT-base	**31.1**	1	**37.4**	7	**43.9**	14	**56.0**	30	**68.9**	48	**78.8**	62

Table 3.1: **Performance comparison**: For each model, we report the accuracy and number of tasks with accuracy above 80%. ind: single-task training, joint: multi-task training, SSL: initialized with self-supervised pretraining on CVR images; IN: pretraining on ImageNet datasets; CLIP: using CLIP's pre-trained vision model.

achieve an increase in performance only when provided with more than 500 training samples. SCL-ResNet-18 performs the best in high data regimes, followed by ResNet-50. SCL and ViT have the lowest performance in high data regimes. This result is unsurprising since transformer architectures generally learn better in high data regimes (millions of data points). This is consistent with prior work [Vaishnav et al., 2022] which finds that ViTs do not learn several SVRT tasks even when trained on 100k samples. Although SCL's performance is near chance, it achieves the best performance when it is augmented with a ResNet-18, which is a strong vision backbone. This jump in performance is indicative of the two architectures' complementary roles in visual reasoning. Results in Table 3.1 and Fig. 3.2 show a clear positive effect of pretraining on all models. SSL pre-trained models achieve the highest performance compared to object recognition and image annotation pretrained models. We observe that ViT benefits from a larger architecture coupled with pre-training on a large image annotation dataset. This highlights transformers' reliance on large model sizes and datasets.

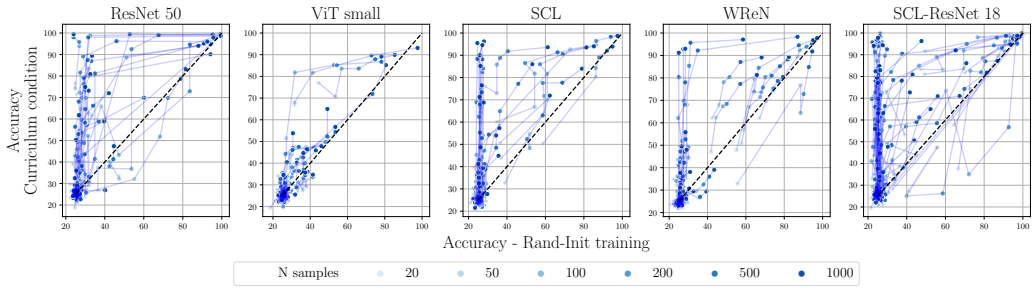
To quantify sample efficiency systematically for all models, we compute the area under the curve (AUC), which corresponds to the unweighted average performance across data regimes. We also introduce the *Sample Efficiency Score* (SES) as an empirical evaluation metric for our experimental setting. It consists of a weighted average of accuracy, where the weights are reversely proportional to the number of samples:

$$SES = \frac{\sum_n a_n w_n}{\sum_n w_n}$$

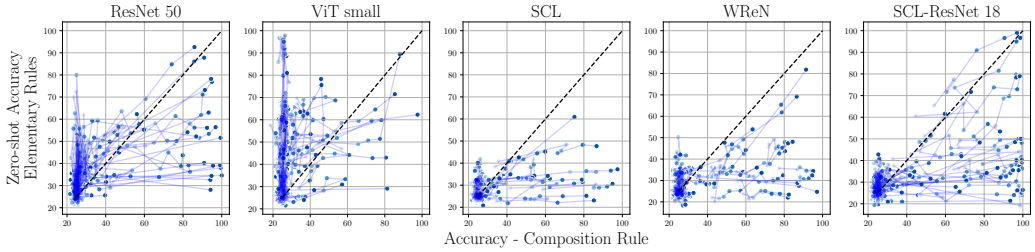
where $w_n = \frac{1}{1+\log(n)}$, n is the number of samples, and a_n is the accuracy at n training samples. This score favors models that learn with the fewest samples while considering consistency in their overall performance. We observe that SCL-ResNet-18 scores the highest in the individual and joint training settings. In the SSL finetuning condition, ViT and ResNet-50 have a similar SES when trained on individual tasks, but ResNet-50 performs better in the joint training setting. These results hint at the efficiency of convolutional architectures in visual reasoning tasks. Collapsing across all data regimes and training paradigms, the best performance on CVR is given by ResNet-50 in the joint training setting with 10k data points per rule. It achieves 93.7% accuracy. This high performance in the 10,000 data regime demonstrates the models' capacity to learn the majority of rules in the dataset and suggests that failure in lower data regimes is explained by their sample inefficiency.

N train samples			SES	AUC
rand-init	ind	ResNet-50	33.7	34.9
		ViT-small	31.3	31.7
		SCL	29.9	30.3
		WReN	33.4	34.1
		SCL-ResNet 18	**38.4**	**39.5**
	joint	ResNet-50	36.0	38.4
		ViT-small	28.4	28.7
		SCL	32.2	33.9
		WReN	30.9	32.0
		SCL-ResNet 18	**37.6**	**40.4**
SSL	ind	ResNet-50	52.4	54.5
		ViT-small	**54.9**	**56.4**
	joint	ResNet-50	**57.0**	**59.6**
		ViT-small	44.7	46.3
IN	joint	ResNet-50	**43.4**	**46.2**
		ViT-small	31.7	32.9
CLIP	joint	ResNet-50	43.7	47.1
		ViT-base	**48.9**	**52.7**

Table 3.2: **SES results**: SES is the Sample Efficiency Score; it favors models with high performance in low data regimes and consistent accuracy across regimes. SES and AUC are computed over the 20–1000 data regimes. The OOD generalization results are provided in the Annex.



(a) Curriculum



(b) Reverse Curriculum

Figure 3.1: **Compositionality**: We evaluate models' capacity to reuse knowledge. (a) Models trained with a curriculum are compared to models trained from scratch. Models trained with a curriculum are overall more sample-efficient. (b) Models trained on compositions are evaluated zero-shot on the respective elementary rules. Models fail overall to generalize from compositions to elementary rules.

Finally, we compare model performance to the human baseline. We observe in Table 3.4 that humans far exceed the accuracy of all models with only 20 samples. This result aligns with previous work on the SVRT dataset [Fleuret et al., 2011] where participants solved similar tasks with less than 20 samples. These results highlight the gap between humans and machines in sample efficiency and emphasize the need to develop more sample-efficient architectures.

3.3.2 Compositionality

Transferring knowledge and skills across tasks is a crucial feature of intelligent systems. With our experimental setup, this can be characterized in several ways. A

N train samples		10000
rand-init	ResNet-50	**93.7 93**
	ViT-small	58.7 37
	SCL	56.9 34
	WReN	64.5 43
	SCL-ResNet 18	78.9 73
SSL	ResNet-50	**93.1 97**
	ViT-small	81.6 67

Table 3.3: **Performance in a high data regime**: We report the accuracy and number of tasks with accuracy above 80%. Models are trained in the multi-task setting.

N training samples	20		1000	
ResNet-50	28.0	0	57.9	14
ViT-small	29.3	1	32.7	3
SCL	26.4	0	44.9	11
WReN	27.5	0	42.4	10
SCL-ResNet 18	26.8	0	**64.1**	**18**
ResNet-50 SSL	45.7	7	**78.3**	**25**
ViT-small SSL	38.7	6	60.3	17
Humans	**78.7**	**26**	-	-

Table 3.4: **Human Baseline**: The performance of models on joint training experiments is compared to the human baseline. The analysis is restricted to the 45 tasks used for evaluating humans. ResNet 50 approaches human-level performance only after SSL pre-training and fine-tuning on all task rules with 1000 samples per rule. Which is 50 times higher than the number of samples needed by humans.

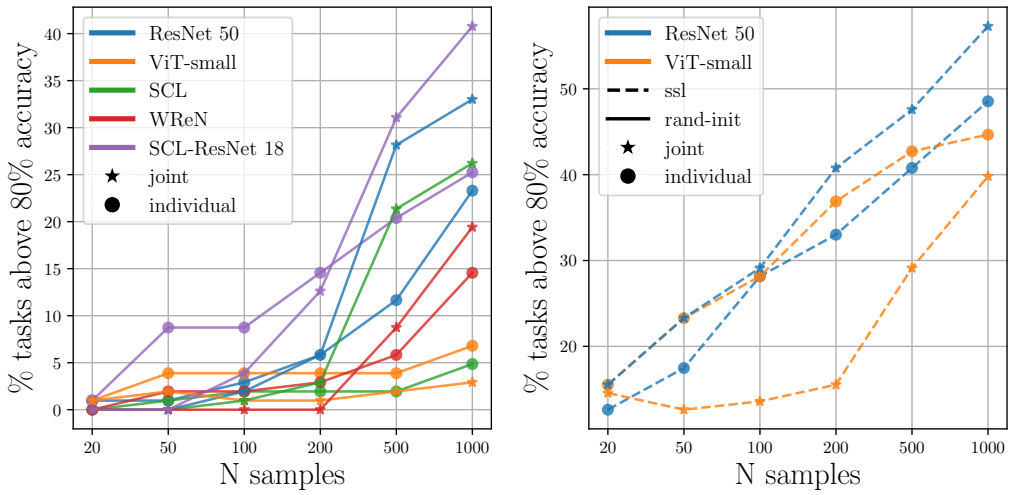


Figure 3.2: **Sample efficiency**: The percentage of tasks for which performance is above 80% plotted against the number of training samples per task rule, with random initialization (left) and SSL pre-training (right).

compositional model should reuse acquired skills to learn efficiently. Thus, when it learns all rules jointly, it could be more sample-efficient because tasks in the dataset share components. In Table3.1 and Figure3.2, we observe that ResNet-50 achieves higher performance on joint training compared to individual rule training, while ViT has the opposite effect. The trend is consistent across data regimes and other settings. These results highlight convolutional architectures' learning efficiency compared to transformer architectures.

We investigate compositionality further by asking whether learning elementary rules provides a good initialization for learning their compositions. For example, a model that can judge object positions and sizes should not require many training samples to associate sizes with positions. We pick a set of complex rules with at least two different elementary relations, train models to reach the maximum accuracy possible on component relations, and then fine-tune the models on the compositions. We call this experimental condition the curriculum condition since the condition is akin to incrementally teaching routines to a model. We compare model performance in the curriculum condition to performance when training from scratch. The results highlighted in Figure 3.1a show positive effects for most models but more significantly for convolution-based architectures. These results indicate that the baselines use skills acquired during pre-training to learn

the composition rules, and that this pre-training helps to varying degrees. We refer readers to the annex for additional analyses and quantitative results.

Finally, we evaluate the transfer-learning from composition rules to elementary rules. We name this condition the reverse curriculum condition. The working hypothesis is that models that rely on compositionality will be able to solve elementary relations without fine-tuning if they learn the composition. We compare performance on a composition rule to zero-shot accuracy on the respective elementary rules in Figure 3.1b. We observe that all models perform worse on elementary relations. These results might indicate that although the baselines could transfer skills from elementary rules to their compositions, they do not necessarily use an efficient strategy that decomposes tasks into their elementary components. Additional analyses are presented in the Annex.

3.3.3 Task difficulty

We analyze the performance of all models in the standard setting: joint training on all rules from random initialization. Figure 3.3 shows the average performance of each model on each elementary rule and composition rule. Since the dataset contains several compositions of each pair of elementary rules, the accuracy shown in each square is averaged over composition rules that share the same pair of elementary rules. Certain rules are solvable by all models, such as the *position*, *size*, *color*, and *count* elementary rules. Additionally, other rules pose a challenge for all models; these rules are compositions of *count*, *flip*, *rotation*, or *shape*. Models that rely on a convolutional backbone were able to solve most spatial rules: *position*, *size*, *inside*, and *contact*. However, they fail on rules that incorporate shapes and their transformations: *shape*, *rotation*, *flip*. Composition rules built with the *Count* relation proved to be a challenge for most models. We believe that models are capable of solving several tasks, such as the *counting* elementary rule, by relying on shortcuts; this could be a summation of all pixels in the image, for example. These shortcuts prevent models from learning abstract rules and hinder generalization. In line with the previous results, SCL-ResNet-18 seems to solve more elementary rules and compositions than the other three models.

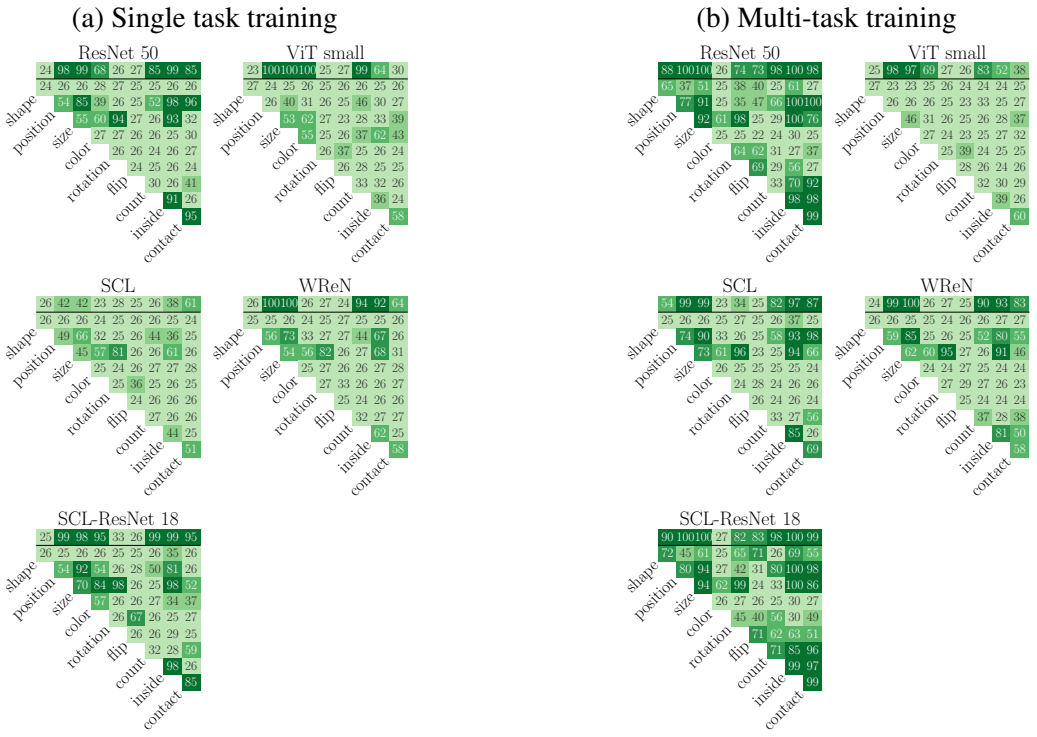


Figure 3.3: **Task analysis**: The performance at 1000 samples is shown for each model. Performance on elementary rules is shown on the top row of each matrix. The elementary relations of each composition are indicated by the annotations. Performance is averaged over different compositions of the same pair. We observe that most models fail on "color"-based tasks.

3.3.4 Out-Of-Distribution Generalization

All baselines are evaluated on the generalization test set of the benchmark. The aim is to determine whether baselines have learned the abstract rules of the tasks and are robust to many variations in the statistics of the input. The generalization testset is created such that it maintains the abstract rules of the task while changing task-irrelevant parameters in the generation process. For example, the shapes have fuzzy contours, unlike the clear lines in the training set. The results in Tables 3.5, 3.6 and Figure 3.4 show that the performance of all models is significantly lower in the generalization test set. For example, the best-performing model, SSL-pretrained ResNet-50, with a SES score of 57% drops to 39%; it

solves 20 tasks instead of 59, with above 80% accuracy in the 1000 data regime. These results are not surprising since deep learning models are known to generalize poorly outside their training distribution. OOD generalization is an important challenge for deep learning models to overcome to reach human intelligence.

N train samples			20		50		100		200		500		1000	
rand-init	ind	ResNet-50	26.3	0	28.1	0	29.1	1	30.3	2	31.5	3	34.3	6
		ViT-small	26.9	0	28.1	0	28.8	1	29.4	2	30.2	3	31.6	2
		SCL	26.0	0	27.8	0	28.0	0	27.9	0	28.5	1	29.7	2
		WReN	27.2	0	28.8	1	29.4	1	30.1	2	31.4	3	32.3	5
		SCL-ResNet-18	**28.8**	**1**	**31.1**	**1**	**31.7**	**3**	**32.4**	**4**	**34.4**	**6**	**38.4**	**11**
	joint	ResNet-50	26.0	0	26.6	0	27.8	1	30.0	1	**37.3**	**7**	**41.3**	**15**
		ViT-small	**26.2**	0	26.4	0	26.6	1	26.9	1	27.4	1	26.9	1
		SCL	25.4	0	25.6	0	27.5	0	30.3	0	33.6	5	35.6	8
		WReN	26.1	0	25.9	0	26.8	0	27.8	0	31.9	4	34.1	6
		SCL-ResNet-18	26.0	0	**27.0**	0	**29.9**	**3**	**32.1**	**4**	34.7	6	37.9	6
SSL	ind	ResNet-50	32.0	5	37.0	6	38.8	**10**	40.9	**10**	42.4	**12**	44.4	**17**
		ViT-small	**36.2**	5	**39.7**	**7**	**40.6**	7	**41.7**	7	**43.2**	9	**45.3**	14
	joint	ResNet-50	34.0	4	**34.3**	4	**37.9**	**8**	**38.4**	6	**46.4**	**15**	**51.0**	**20**
		ViT-small	**34.1**	**6**	33.0	**5**	32.5	4	33.2	6	33.4	6	35.9	11

Table 3.5: **OOD Generalization Results**: Models perform significantly worse on the generalization test set.

3.4 Discussion and Future Work

In this work, we have focused on two important aspects of human intelligence—compositionality and sample efficiency—that are scarcely addressed in the evaluation of deep learning models. Using the CVR benchmark, we performed an analysis of the sample efficiency of existing machine learning models and their ability to harness compositionality. Our results suggest that even the best pre-trained neural architectures require orders of magnitude more training samples than humans to reach the same level of accuracy, which is consistent with prior work on sample efficiency [Lake et al., 2015]. Our evaluation further revealed that current neural architectures fail to learn several tasks even when provided with an abundance of samples and extensive prior visual experience. These results highlight the importance of developing more data-efficient and vision-oriented neural architectures for achieving

N train samples			SES	AUC
rand-init	ind	ResNet-50	29.4	29.9
		ViT-small	28.8	29.2
		SCL	27.7	28.0
		WReN	29.5	29.9
		SCL-ResNet-18	**32.2**	**32.8**
	joint	ResNet-50	**30.3**	**31.5**
		ViT-small	26.6	26.7
		SCL	28.8	29.6
		WReN	28.1	28.8
		SCL-ResNet-18	**30.3**	31.3
SSL	ind	ResNet-50	38.3	39.2
		ViT-small	**40.5**	**41.1**
	joint	ResNet-50	**39.0**	**40.3**
		ViT-small	33.6	33.7

Table 3.6: **Out-Of-Distribution Generalization SES Results.**

human-level artificial intelligence. In addition, we evaluated models' generalization ability across rules, from elementary rules to compositions and vice versa. We find that convolutional architectures benefit from learning all visual reasoning tasks jointly and transferring skills learned during training on elementary rules. However, they also failed to generalize systematically from compositions to their individual rules. These results indicate that convolutional architectures are capable of transferring skills across tasks but do not learn by decomposing a visual task into its elementary components. The poor sample efficiency and generalization of neural networks compared to humans could be due to their non-compositional learning strategy and lack of curricula in their training. This idea is supported by behavioral and computational evidence [Dekker et al., 2022] where humans are shown to generalize compositionally beyond the capacities of neural networks. Furthermore, they benefit from curricular training, which highlights the importance of introducing curricula to the training tasks.

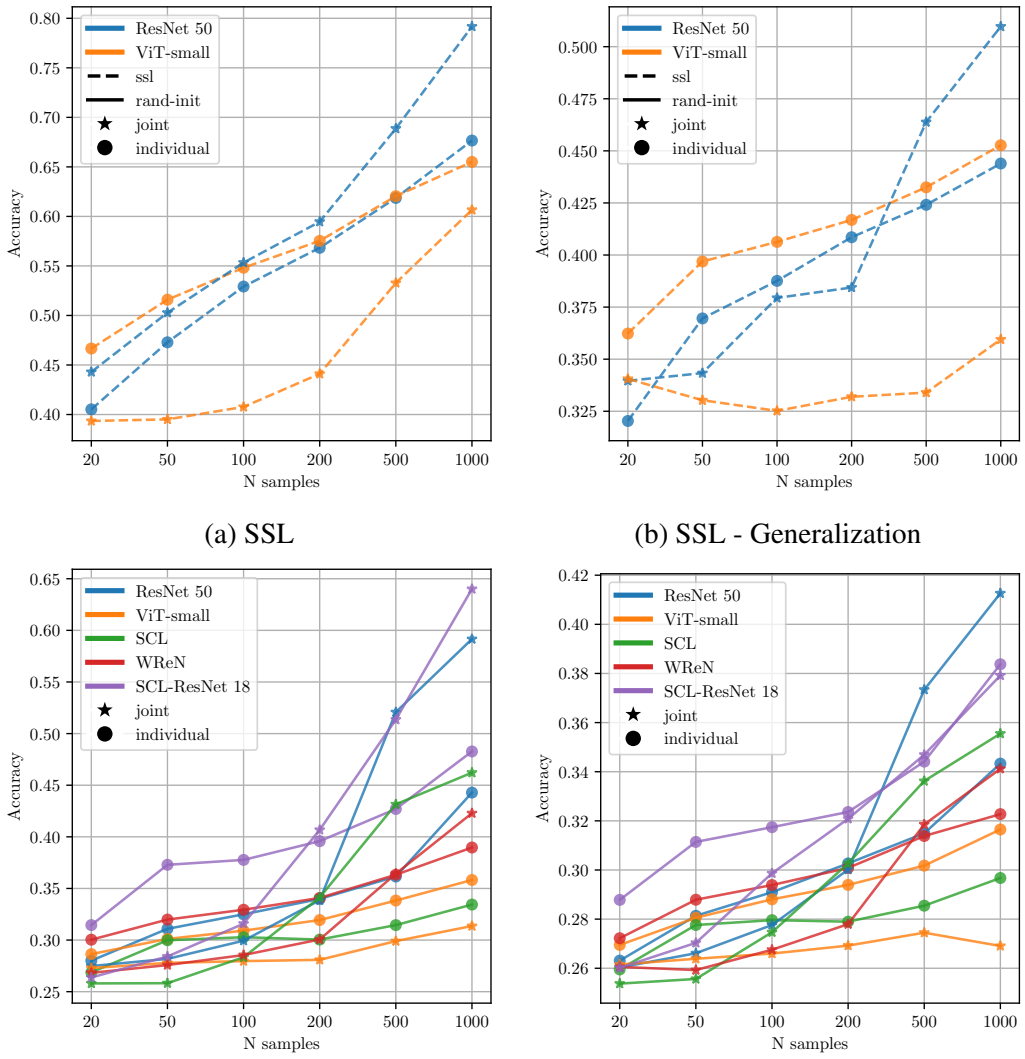
While our work addresses important questions on sample efficiency and compositionality, our evaluation methods could be further improved and adapted to different settings. For example, the sample efficiency score is an empirical metric used only for evaluating our benchmark. It requires training all models on all data regimes for the score to be consistent. Although our work is not unique in ad-

dressing sample efficiency, it aims to promote more sample-efficient and general models.

3.4.1 Model Design For Sample Efficiency And Compositionality

In the visual reasoning literature, general-purpose models such as ViTs and CNNs are provided as baselines, with more complex approaches relying on additional inductive biases for reasoning such as RNNs, GNNs, and Relation Networks [Johnson et al., 2017a, Santoro et al., 2017, Chen et al., 2021b]. These architectures achieve decent performance but have poor generalization and sample efficiency. More promising solutions for visual reasoning leverage modularity [Andreas et al., 2016b, Chen et al., 2021c, Hudson and Manning, 2018, 2019, Mittal et al., 2021, Rahaman et al., 2021, Goyal et al., 2019]. Modular neural networks are composed of a set of modules that perform different operations. These models are generally orchestrated by a controller module that executes language-based instructions. We believe that modularity is a promising inductive bias for developing models that implement compositionality. When equipped with a proper controller module and information routing mechanisms, a modular network could flexibly manipulate novel concepts and build contextual representations. Although these models have the advantages of interpretability and better OOD generalization, they are notoriously difficult to train. Other methods focus on scene understanding [Burgess et al., 2019, Engelcke et al., 2019, Li et al., 2020]; these models rely on attention and object-centered representations as inductive biases for building scene representations, which are useful for visual reasoning [Ding et al., 2021]. In another vein, certain approaches scale up simple architectures based on transformers and convolutions and rely on self-supervised pretraining to achieve impressive performance on several multi-modal computer vision tasks [Ramesh et al., 2022, Yu et al., 2022]. However, the capacity of these models to leverage compositionality is limited by their architectural components: transformers and ResNets. We believe that modularity, attention, and objectness are essential inductive biases to achieve sample efficiency and compositionality in CVR. Attention is used for extracting the scene graph from the image, while the modules implement various strategies to solve different visual reasoning tasks. We believe that future models of visual reasoning should implement these inductive biases while taking inspiration from human cognition in orchestrating visual reasoning as program

execution.



(a) SSL

(b) SSL - Generalization

(c) Rand-Init

(d) Rand-Init - Generalization

Figure 3.4: **Performance across settings**. The accuracy is aggregated over all tasks. Random choice accuracy is 0.25.

CHAPTER 4

PRINCIPLES OF NEURAL ARCHITECTURE DESIGN FOR ACHIEVING HUMAN INTELLIGENCE

4.1 Introduction

The preceding chapters have centered primarily on characterizing different facets of intelligence, highlighting the significance of compositionality as a computational paradigm for efficient learning and generalization, and investigating the disparities between machine and human intelligence in visual reasoning. To narrow the gap between humans and machines in terms of intelligence, a promising approach is to take inspiration from the brain for building AI systems. In line with this view, the objective of this chapter is to introduce principles for architectural design and training strategies that draw inspiration from brain function.

While taking inspiration from the brain is valuable, it is essential to strike a balance between replicating brain functions and recognizing the inherent differences between biological and artificial systems. I believe that replicating every intricate detail of the brain, from the dynamics of neural firing to the anatomical structure, in neural network models might not be necessary for achieving intelligence. Attempting to create an accurate model of the brain can often lead to unnecessary complexity and computational inefficiencies. Instead, my focus lies on understanding the fundamental principles and mechanisms that contribute to intelligence. By distilling these principles in AI systems, even if they do not exactly mimic brain function, they may exhibit intelligent behavior without being burdened by the complexities of brain function. Identifying the crucial factors for intelligence is a non-trivial task. For instance, the question of whether neurons and their dynamics must be implemented accurately or if different computational units can capture their expressivity remains open. Additionally, discerning which properties emerge from the system and which are innate presents further challenges. For example, in the first chapter, compositionality is hypothesized to be an emergent property of the brain. As such, I believe that for a neural network-based system to effectively implement and leverage compositionality, akin to humans, it cannot rely on a single inductive bias. Instead, the system should learn to use compositionality as a computational paradigm through experience.

Given the priors and emergent properties of human intelligence, I propose a framework that regroups design principles and training schemes for brain-inspired neural network architectures. While this work involves several ideas from cognitive science and compiles them into a set of design principles, it is important to acknowledge that they may not be exhaustive given our limited understanding of brain function. Moreover, the implementation proposed for these principles may

pose significant technical challenges. As an initial version, this framework can be improved as research on cognitive and neuroscience progresses.

To provide proof of concept, I developed a neural network model following this framework: AbstractNet, a modular architecture that can control and adapt computations to task demands. Preliminary experiments AbstractNet shows its capacity to solve many tasks involving various skills by learning to manipulate many modules in an end-to-end fashion. To conclude, I will discuss potential avenues for improving this framework.

4.2 Taking inspiration from brain function

The first chapter characterizes the intelligence of a system by its performance and efficiency in using its capacities to interact with its environment and solve tasks. Assuming the system's adequate understanding of the task, its intelligence can be evaluated by four factors:

- **Performance**: Its success at solving the task or providing a detailed and accurate procedure for solving the task.
- **Time efficiency**: The average computation and execution time required by the system for solving an instance of the task.
- **Energy efficiency**: The number of computations required by the system for solving an instance of the task, assuming an equal energetic cost of a computation unit.
- **Data efficiency**: The number of distinct problem instances required for a system to reach its maximum performance level.

The brain excels at these factors due to several innate and emergent properties. The innate properties include 1) *biological priors*; the architectural organization of brain areas that is optimized for processing specific signals; 2) *learning*; the ability to partially change its structure and connections between neurons to store information and adapt behavior; and 3) *agency*; the capacity to interact with the environment and control parts of it to perform complex tasks. Biological priors honed throughout evolution provide the system with an initialization that can be

adapted through experience for optimal behavior through learning. Agency allows the system to interact with the environment to acquire experience, which is used for learning. Focusing on biological priors, modularity can be observed as a salient feature of the brain. The brain is comprised of many parts with distinct functional roles. Within the neocortex, a critical brain region for learning, cortical areas can be partially distinguished based on their cytoarchitecture, i.e., the types of neurons and their local patterns of connectivity. Their functions depend on the signals that they process, which can be inferred from the neighboring areas, regions, and organs that they are connected to. For example, the occipital lobe processes mainly visual information since it receives input from the retina. It follows that cortical areas can be repurposed for different functions if they are not recruited for their biological function. For example, the visual cortex of people who have lost their vision is active during tasks such as braille reading, auditory processing of words, or sensory discrimination of tactile stimuli [Burton, 2003]. While the brain undergoes such structural changes over long periods, it still demonstrates its flexibility and adaptability. Many neuroscientific studies investigate networks within the brain. For example, among many prominent cognitive science theories about consciousness and brain function, the global workspace theory [Baars, 1988] stipulates that the various modules of the brain interact partly through the coordination of a global workspace. This theory has inspired interesting ideas for building cognitive architectures that we will discuss later [VanRullen and Kanai, 2021, Goyal et al., 2022].

Another biological prior is the capacity of neurons and neural populations to perform many powerful computations, such as storing patterns of activation over long periods (memory), processing and filtering information based on its source and content (attention), and parameterizing computation based on context (recurrence and dynamic parametrization). The formation of memories and the learning of new computations in neural populations are mediated through various forms of neuroplasticity. Lasting changes in neuronal activity result from various learning rules, including reward-driven learning, error-based learning, and learning based on self-organization. Different brain regions rely on different learning rules; for example, while the hippocampus is associated with a pattern completion learning rule, the basal ganglia relies primarily on reward-driven learning, which has been modeled in reinforcement learning, and the cerebellum is associated with error-based learning. Interestingly, neurons of the neocortex rely to varying degrees on these three learning rules. The rich variety of connectivity patterns and learning dynamics in the brain is important for supporting various

cognitive functions [Atallah et al., 2004]. Among the theories that attempt to explain the complicated learning rules in the brain, the Complementary Learning Systems theory [McClelland et al., 1995, Kumaran et al., 2016] propose that phenomena of short-term and long-term memory are supported by fast and short-term learning in hippocampal neurons cortical that aids slow and long-term learning in cortical neurons. In this theory, these complementary systems allow the brain to flexibly learn new concepts while maintaining a stable structure in the neocortex. Although the field of AI favors applying unique inductive biases on a broad scale in modern systems for greater performance, the low efficiency of these models may be due to the limited inductive biases that they employ. This theory gives ideas on how involving different types of learning rules and architectures could potentially solve the rigidity and poor generalization issues of AI systems.

There's an important role for experience in shaping this blank slate into a highly intelligent system. The brain learns from interactions with its environments which provide unlimited *continuous*, *sequential*, and *multi-modal* inputs. Within this environment, the brain is confronted with a variety of tasks that vary in relevance to itself. Another important characteristic of experience is its inherent *compositionality* and redundancy.

From the interactions between the brain and its environment emerge complex architectural, computational, and functional phenomena that contribute to and characterize intelligence. Among the emerging architectural phenomena is the hierarchical organization of cortical areas according to function and the abstraction of information. Due to this hierarchy, computations are arranged as *pathways* composed of cortical areas, an example of which is the ventral visual pathway that processes visual information to extract information about scene components such as object categories and features. At the apex of this hierarchy, a *cognitive control* system emerged to coordinate computations in several brain areas. This cognitive control system uses bidirectional connections with several cortical areas to receive multi-modal information and exert top-down control to route information through relevant pathways based on task demands. Furthermore, this cognitive control system plays an important role in learning complex behaviors. Due to pressure for efficiency and optimal behavior, the cognitive control system implements abstraction and compositionality as general computational paradigms. Botvinick and Cohen [2014] discuss these ideas, offer a perspective on the cognitive control system, its function, and its emergence, and highlight its importance in cognitive architecture.

The classical example that behaviorally demonstrates cognitive control is the Stroop task [Stroop, 1935]; when presented with the word "red" written with a green color, reading the word is faster than naming the color in which it is displayed and naming the color is faster when the two colors are the same. This is because humans are biased to unconsciously read words more than to name the colors of the words. To name the color, cognitive control intervenes by inhibiting the semantic information of the word and facilitating the color information routed to the verbal production process. This type of control was modeled as the parametrization of task-relevant neural circuits [Cohen et al., 1990, Cooper and Shallice, 2000, Dayan, 2007, Dehaene and Changeux, 1997, Shenhav et al., 2013] using a gating mechanism that modulates neural activity based on task representations. Dynamic gating of neural activity in general can be implemented as a parametrized multiplicative or additive effect on activity but its function is not limited to filtering and routing information, it could potentially be used for briefly instantiating new computations in neural circuits and instantiate structure in other cortical areas. PFC, as an area that is heavily involved in executive functions, has been shown to implement these control mechanisms by maintaining and manipulating information in working memory and exerting top-down attentional control over other brain regions. The regulation of control pertains to the allocation of control based on the goals and resources of the brain. Humans are capable of coordinating a few tasks simultaneously, however, they can be cognitively engaged in only one task at a time. For example, it is possible to wash the dishes while talking to someone but only one of these tasks will be cognitively engaging. This explains why humans cannot perform two tasks simultaneously if they require the same resources. For example, it is not possible to read an article while talking to someone and simultaneously understand both. Furthermore, a task where one lacks proficiency will require more cognitive engagement than other tasks. Thus, the brain leverages complex mechanisms to allocate control appropriately and switch between tasks. Several models have been proposed to explain cognitive control allocation and they involve several factors; decisions about engagement in a task could depend on the potential benefits, the risks of failure, and the costs of the control including its intensity [Shenhav et al., 2013], and while performing the task errors or uncertainties encountered can be triggers for the engagement of control. The limits of cognitive control are intuitively explained by the availability of representations in the neural circuits in the examples that we presented. However, it is a phenomenon with many intricacies, a recent review of this literature [Musslick, 2021] proposes an explanation for the limits in capacity in cognitive control in the context of learning and inference.

A full understanding of cognitive control and brain function, in general, cannot be reached without an understanding of how the skills of allocating and applying control are themselves learned by the brain. Thus far, we have seen that these processes involve several inferences over the benefits and costs of performing the task, task representations, and parametrizations of other neural circuits based on these representations. Learning such complex interactions is a tremendous challenge that every developed human brain has surmounted. Botvinick and Cohen [2014] propose that, similarly to how the visual cortex is shaped by the statistics of the visual experience, the brain's cognitive control system is shaped by the statistics of the space of tasks that it encounters. While the visual cortex learns to represent visual information for behaviorally relevant tasks, the cognitive control system learns how to coordinate cortical areas for optimal behavior by flexibly and efficiently representing and performing a wide range of tasks. Interestingly, control was found to implicate factorization [Rougier et al., 2005] and structure inference [Collins and Frank, 2013] of task rules and features in biologically plausible models. In Collins and Frank [2013], participants spontaneously inferred a task structure without being instructed. This suggests structural decomposition of information is a bias that experience is embedded in cognitive control systems.

While the precise mechanisms and neural circuits involved are still a topic of ongoing research, several key brain regions are known to play important roles in implementing cognitive control processes. These regions include the PFC, anterior cingulate cortex (ACC), parietal cortex, basal ganglia [O'Reilly and Frank, 2006a], and brainstem [Aston-Jones and Cohen, 2005, Braver and Cohen, 2000]. PFC regions, particularly the dorsolateral prefrontal cortex (dlPFC) [Miller and Cohen, 2001a], and in part medial frontal and superior parietal cortex [Duncan, 2010, Duncan and Owen, 2000], are crucial for exerting cognitive control. Various models try to explain mechanisms of cognitive control [Cohen et al., 1990, Miller and Cohen, 2001b, Anderson et al., 2004, O'Reilly and Frank, 2006a, Koechlin and Summerfield, 2007], and these models might not paint the full picture of the neural basis of cognitive control as other studies propose the existence of two distinct networks specialized in cognitive control [Dosenbach et al., 2008] and propose a role for the insula in control and attention [Menon and Uddin, 2010], but there is an overall agreement on the role of PFC in exerting and allocating control on several cortical areas through top-down connections, the hierarchical organization of control in the PFC, the role of ACC in modulatory feedback mechanisms and regulation of PFC-mediated control, the role of basal ganglia in the dynamic gating of information in the PFC and learning. PFC's role in cognitive control ex-

plains its involvement in functions that we discussed above; learning, abstraction, structured representations, and building world models.

Zeithamova et al. [2019] review studies on the mechanisms of concept learning. These studies highlight the involvement of the hippocampus (HPC), ventromedial PFC (vmPFC), rostolateral PFC (rlPFC), and lateral PFC (lPFC) in concept learning and generalization. The findings point to the cooperation of vmPFC and HPC during early learning where HPC maintains and updates specific and generalized memories of concepts and vmPFC leverages attention to focus on relevant features and ignore irrelevant ones [Mack et al., 2016, Constantinescu et al., 2016, Mack et al., 2020].

Other studies show the involvement of PFC in reasoning and inference. Rostolateral PFC (also known as anterior PFC and frontopolar PFC) supports relational reasoning [Christoff and Gabrieli, 2000, Christoff et al., 2009], abstraction [Bunge et al., 2003, Christoff et al., 2001], prospective memory [Gilbert, 2011, Momennejad and Haynes, 2012, 2013] and other processes such as analogy and problem solving [Christoff et al., 2001, Kroger et al., 2002, Bunge et al., 2003, 2005, Green et al., 2006, Hampshire et al., 2011, Watson and Chatterjee, 2012]. Interestingly, complex problems such as Raven's Progressive Matrices (RPM) activate several PFC regions especially when multiple relations must be combined before finding an answer in contrast to problems with a single underlying relation [Christoff et al., 2001, Kroger et al., 2002]. However, other cortical areas are involved in reasoning in specific contexts such as physical simulation [Ahuja et al., 2021] and conceptual combination in language [Frankland and Greene, 2020].

Structured representations of information in the brain exist in various cortical regions but prefrontal medial temporal lobe areas are of particular interest since they are hypothesized to be involved in the structured representation of abstract concepts [Behrens et al., 2018]. Manns and Eichenbaum [2006] hypothesize that HPC models a conjunctive representation between sensory representations transmitted by the lateral entorhinal cortex LEC and the structure embedded in the medial entorhinal cortex MEC. This model was further elaborated [Whittington et al., 2020] and used for explaining the discovery of grid cells in MEC and place cells in HPC among a variety of other cell types. Other studies that build on these ideas and suggest that mPFC and HPC play complementary roles in generalization [Samborska et al., 2022]; while mPFC maintains task structure across problem instances, HPC remaps the sensory information of each problem within the same structure. Theves et al. [2021] study the interactions between these ar-

eas while learning the structure of problems and propose that mPFC and HPC integrate evidence accumulated from sensory experience to update hierarchical concept representations in rlPFC.

The brain's capacity to build models of its environment and simulate scenarios within it is hypothesized to involve modality-specific cortical areas. For example, certain motor areas are thought to be involved in simulations over spatial and temporal predictions [Schubotz, 2007], as in mental rotation [Zacks, 2008] and physical simulation [Fischer et al., 2016, Battaglia et al., 2013]. The theory about brains leveraging models of the environment to plan action sequences to reach future goals has been supported by many theoretical and behavioral studies [Daw et al., 2005, Dickinson and Balleine, 2002, Dolan and Dayan, 2013, Schoenbaum et al., 2009, Tolman, 1948]. Importantly, these models can have transition structures at different levels of abstraction and timescales. For example, "traveling from New York to Paris" can be hierarchically decomposed into a sequence of actions from "going to the airport", "going to your car" and "standing up from your desk". The brain can reason at any level of granularity and simulate state transitions at various timescales. Several studies explain these concepts in the context of hierarchical reinforcement learning [Botvinick and Plaut, 2009, Badre et al., 2010, Badre and Frank, 2012, Gershman et al., 2015, Balaguer et al., 2016] and propose a role for PFC and basal ganglia in planning and executing behavior hierarchically.

In light of this large body of work, the brain can be viewed as a modular system where activity is coordinated by a central module considered a controller. By activating specific modules and routing information between them, the system can form pathways to implement the complex high-level functions that we discussed earlier. These ideas are the basis of the framework that we propose and inspire the architecture developed in later sections.

The benefits of abstraction and compositionality have been discussed in Chapter 1. They support several complex functions as efficient solutions for learning compositional tasks and performing fast inference while minimizing errors. These functions include meta-learning, building world models and simulating interactions within them, planning, balancing exploration and exploitation in new environments, and many other strategies. Meta-learning, as the name implies, is a skill by which the system becomes more efficient at learning new tasks through transfer from prior tasks and by learning to select task-relevant information to build a model of the task. Planning and world model simulation support learning and

play an important role in efficient inference and generalization.

Although we classify several high-level functions as emergent, the choice of architecture remains paramount since these functions could not emerge if the architecture and the learning process did not promote their emergence. Simulation, for example, relies on the brain's capacity to learn transition structures in the environment, which is facilitated by the predictive learning rule. The predictive learning hypothesis claims that the brain is constantly predicting future outcomes at various timescales and learns by contrasting predictions with outcomes. Learning through future prediction allows the brain to model the environment and generate outcomes based on imagined scenarios. Another example is the visual cortex and its main pathways; the connectivity of their neural circuits facilitates performing a wide variety of visual tasks such as object recognition, search, and physical reasoning. Other pathways might exist for model building and other high-level functions that are important for intelligence. Nevertheless, if such pathways do not exist, the brain has the adaptive capacity to develop them.

Overall, isolating the characteristics of brain function that contribute to its intelligence remains a challenge. However, we can identify key principles of its construction that contribute to important high-level functions: distinction of processing components, variety of architecture and specialization in each component, coordination of function by an executive system within and across components, adaptability and control over learning, agency, and a structured and varied learning experience.

4.3 Adapting principles to machines

The computational frameworks that have been used for simulating brains range in the level of detail that they model, from models of neural dynamics that simulate the physical equations of neural spiking to statistical models that simulate the behavior of cortical areas at a high level in specific tasks. Beyond matters of technical feasibility, practicality, and fidelity of modeling brain function, it is important to ponder the level of detail necessary or sufficient for achieving human-level intelligence on a machine. Spiking neurons might contribute to the system's efficiency, but they might not be crucial for achieving intelligence if their transmission of information and plasticity can be performed by an equivalent system on machines. Artificial neurons, which are simplified models of their biologi-

cal counterparts, are the computational units that we use for implementing these principles. They present the best trade-off between fidelity and technical feasibility, as shown by the exponential progress in AI research during the last decade. Since their success, there have been many attempts to develop more biologically plausible neural networks, and the principles that we describe above have been addressed in several studies. We aim to offer a different perspective on their application in artificial neurons that delves into the technical difficulties and advantages of combining these principles.

To account for the separation of processing components and their variety, the neural architecture can include several modules with varying inductive biases. While certain modules are specialized for sensory inputs and action outputs, other modules have the role of abstract computations that can be leveraged by the system for learning general skills. One particular module, which accounts for executive function, determines which modules to use and the routing of information within the architecture. It also determines the dynamic parameterizations of other modules, which can implement attention, among other mechanisms. The neural architecture is recurrent; it runs several steps, at each step selecting modules to run, the flow of information between the modules, inputs to process, and actions to output. These aspects of architecture partially account for its agency since it does not control only the actions it performs but also its internal computations. To be fully autonomous, the system must be capable of computing rewards and error signals, then attributing learning signals to the relevant modules and updating them. The learning experience is difficult to account for due to its high complexity, uniqueness to individuals, and open-mindedness. However, certain aspects can be accounted for: the variety of tasks, the multi-modality of inputs, the inherent compositionality of tasks and inputs, the gradual increase in complexity, and a training setup that relaxes computation time constraints on the system.

In sum, the principles of architecture design and training consist of:

- Modularity, recurrence, and variety of inductive biases across modules.
- Control over internal computations, module activation, and interactions between modules.
- Control interactions with tasks and unconstrained use of time.
- Partial control over learning rules and credit assignment.

- Diverse tasks, curriculum organization, and compositionality in the learning experience.

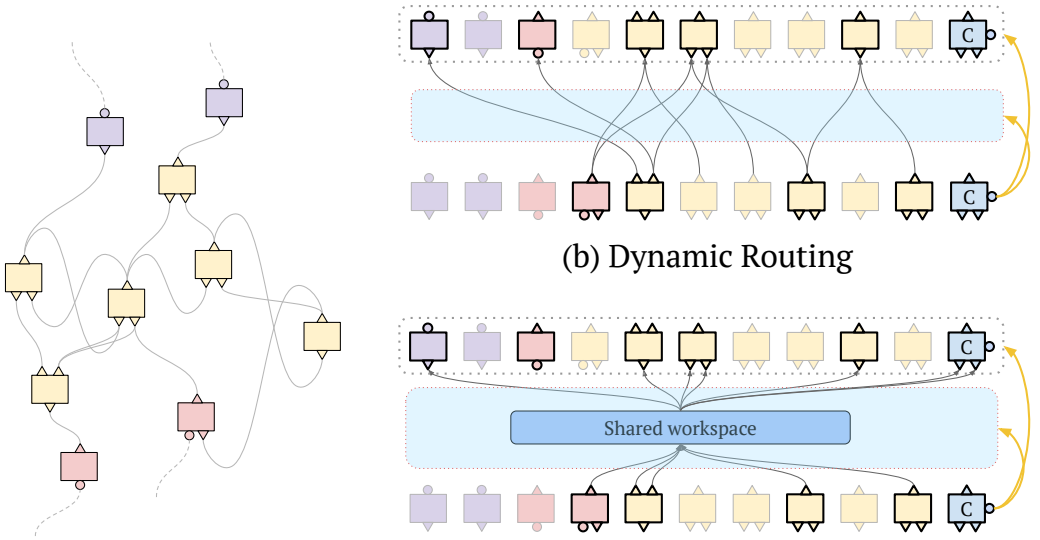
4.4 Implementing design principles

Several implementation choices for the design principles can be adopted in the architecture. In this section, we review a few implementation ideas and discuss their advantages and drawbacks.

4.4.1 Innate properties

Modularity Modularity was an early topic in research on neural networks [Auda and Kamel, 1999] and has seen significant advances following progress in deep learning [Andreas et al., 2016b, Kirsch et al., 2018, Rosenbaum et al., 2017, 2019, Chen et al., 2020]. While a standard neural architecture consists of a fixed sequence of parametrized functions, layers of neural networks, and activation functions, modular architectures consist of a set of modules m_i for $i \in \{1, ..., N\}$, each module having a different set of parameters. In modular architectures, information goes through a set of successive processing steps; during each step, information is routed to modules based on the design of the architecture. Several aspects of the architecture can be taken into account: the variety of module architectures, the activation of modules at each processing step, connectivity between modules, and the modules' interface with task inputs and outputs.

Inductive biases Considering architectural variety, while some works [Rosenbaum et al., 2017, 2019, Kirsch et al., 2018, Goyal et al., 2019] experiment with sets of simple homogeneous modules that are adapted for the task (convolutions, linear transformations, or recurrent units including activation functions), others experiment with heterogeneous modules [Andreas et al., 2016b, Chen et al., 2020] especially in visual reasoning tasks. In the context of multi-task learning, modules that have similar inductive biases are in general suitable for homogeneous tasks—exclusively visual tasks, for example—but could be inefficient when solving heterogeneous tasks. For example, visual question answering is a complex task that requires visual processing, language understanding, abstract reasoning, attention, and memory. To support many complex functions efficiently, the ar-



(a) Fixed Routing

(b) Dynamic Routing

(c) Shared Workspace-based Routing

Figure 4.1: **Modularity and Routing**. a) In fixed connectivity, connections between modules are built into the architecture. b) In dynamic routing, information is routed to other modules based on the specification of the controller; the input to each input gate is an aggregate sum of outputs from other modules. c) In the shared workspace routing, information from different modules is integrated into a large embedding vector. The same vector is used for creating inputs for other modules. Shared workspace routing is less constrained than dynamic routing.

chitecture could require a variety of specialized modules. A visual module can be based on convolutional or attention-based architectures, and a memory module can include a memory control architecture or Hebbian learning-based attractor networks. Other architectures can be based on task-general inductive biases such as recurrence (RNNs), gating (LSTM and GRU), attention (transformers), and dynamic parametrization (hypernetworks).

Top-down modulation Another interesting design choice for the modular architecture is top-down control. It allows the system to adapt the module's computation for the task. For example, top-down control in a visual module can be used for deploying spatial or feature-based attention. It can also be considered a more special case of dynamic parametrization, similar to gating mechanisms.

Implementing useful top-down control mechanisms depends on the module's architecture; it can involve a parallel architecture with outputs at several steps of processing in the module. Various forms of control are explained in Figure 4.2 and Algorithm 3.

Module activation The activation of modules at processing steps can be fixed based on the model's architecture or determined during inference. For example, the standard choice is to activate all modules at each processing step or to assign sets of modules for specific processing stages, akin to a multi-layer architecture. Alternatively, the model could rely on a learning process for selecting modules to activate in one or multiple steps. For example, a special module, named the controller [Kirsch et al., 2018] or the router [Rosenbaum et al., 2017], trained with reinforcement learning would select one or many modules to activate at each processing step. Module activation can also be determined through bottom-up competition between modules [Goyal et al., 2022]. When considering heterogeneous modules and the potential implementation of top-down control in each module, it's possible for one module to afford many computations $a_j^{m_i}$ for $j \in \{1, ..., A_{m_i}\}$. For example, a memory module can read from memory or write to memory, and a visual module can process information in a bottom-up fashion to extract features or use top-down signals to attend to specific information in the input. From this perspective, it could be useful for the control scheme to activate computations rather than modules.

Routing Information routing in a modular architecture can be implemented in several ways. Considering that a set of modules is assigned for processing task inputs and providing outputs while other modules process information coming from other modules, we can refer to module input or output gates $g_j^{m_i}$ for $j \in \{1, ..., G_{m_i}\}$ as internal if they interface with other modules and external if they interface with task variables. Since the connections of external gates with task variables are fixed, we are interested in the connectivity between internal input and output gates. This connectivity can also be fixed, rendering the architecture fixed as in a standard neural network, or it can be specified based on a routing scheme. An input gate can receive one output vector as in Rosenbaum et al. [2017] or a combination of many output vectors from different modules as in Kirsch et al. [2018]. In the second case, the combination of output vectors depends on the routing scheme. Simple solutions include a summation [Kirsch et al., 2018] or a weighted average of input vectors using softmax normalized weights predicted by a routing function (the controller, for example). In this design, the dimensionality

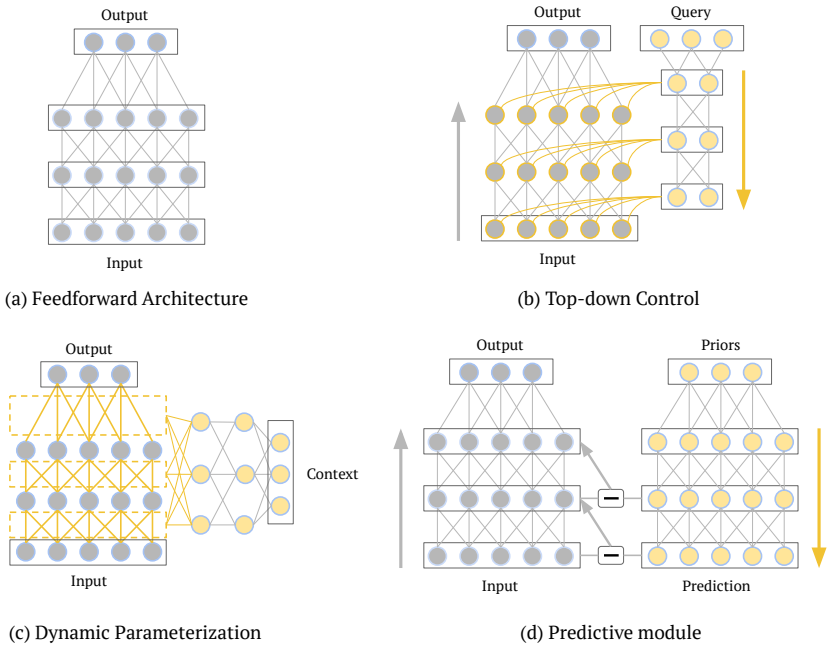


Figure 4.2: **Control Architectures**. a) A standard neural network processes information in a feedforward manner; it has no form of control. b) Top-down modulation of activity can be achieved through multiplication or addition of the intermediate based on top-down activations. c) In dynamic parametrization, a module generates the weights that are used for processing inputs. d) Predictive learning model: a parallel module predicts the inputs, contrasts them, and sends error signals to the main module or other modules.

of internal gates is fixed across modules, and the modules share an input and output representation space. The latter constraint might limit the function of the information content expressed by different modules. To avoid this constraint, it's possible to add an intermediate transformation of the module output vectors based on embeddings of the source and target modules. This process can be considered a translation between module representational spaces. Another advantage of this approach is that it supplies information about the source of the input vector and adapts the distribution of the output to the distribution of gate inputs. An example of a more sophisticated non-linear combination of vectors is the shared workspace representation [Goyal et al., 2022]. In this routing scheme, information is written into a shared representation at one step of processing, which is transmitted to all active modules at the next step. Similar to module or action activation, routing

can be determined by a controller, possibly the same that activates modules, by specifying a routing matrix at each processing step or specifying write and read vectors from the shared workspace for each module. Alternatively, the modules can route information through bottom-up attention mechanisms such as query-key-value (transformer-based) attention, as in RIMs [Goyal et al., 2019, 2022]. Different routing methods are explained in Figure 4.1.

Controller design In this paper, we will focus on controller-based module activation and information routing. The controller is an inspiration from the cognitive control system in the brain, which is responsible for coordinating computation in cortical areas through top-down control. Early recurrent architectures that incorporate gating, such as LSTM [Hochreiter and Schmidhuber, 1997], have been used as models of the prefrontal cortex and its role in cognitive control, among other complex cognitive functions [Wang et al., 2018, O'Reilly and Frank, 2006b]. Gated RNNs have also been used for controlling memory access and manipulation in several models [Graves et al., 2014, 2016, Wayne et al., 2018]. Recurrence and gating are mechanisms that allow for the flexible manipulation of information and maintenance of working memory over long timespans. This has been demonstrated in a variety of tasks. A relevant example is the capacity of LSTMs to implement a learning algorithm in their dynamics, allowing them to perform meta-reinforcement learning [Duan et al., 2016]. Gated RNNs seem suitable for the control of internal computations in a modular architecture. The specification of the input to the controller is also an important design choice. The controller should have access to the embedding of task inputs. Furthermore, having access to intermediate module outputs allows the controller to monitor execution. In a multi-task setting, the model can learn task embeddings or have access to task embeddings provided with the curriculum. Other useful information for the controller is the internal state of the model, previously active modules, and routing decisions. The output of the controller can be used for direct top-down control over specific modules. In this implementation, the controller can be considered a special module that is active in all processing steps and has the added role of deciding module activations and routing.

Adaptive computation time Another important aspect of the architecture is the management of its interactions with task instances. The architectures that we have reviewed thus far either fixes several computation steps akin to layers of processing in a deep network or adapt them to the input size when the data is sequential, as in Routing RNNs [Cases et al., 2019]. An alternative approach is to allow

the model to allocate its computation time by deciding when to read inputs and emit outputs. Prior work [Graves et al., 2016, Banino et al., 2021] implements adaptive computation time with scalar halting probability at each computation step. A similar approach can be implemented by including task interactions such as reading inputs and task termination as actions decided by the controller similar to the activations of modules. The halting probability is replaced by actions in task interactions. The controller can be fed additional task information about the task state, such as a pending input or an expected output, to guide its decisions. Learning to allocate computation time has the potential benefit of allowing the model to adapt inferences to task requirements. For example, it can learn efficient programs for simple tasks and generalize them to more complex tasks by allocating more computation time.

Predictor modules Beyond modular architecture, predictive learning is another biologically inspired design principle that has gained popularity in AI research, especially in applications such as self-supervised or unsupervised learning. The general idea is to learn to predict unknowns based on the available information. This approach has been used for pretraining vision and language models on large, unlabeled datasets. It's also the basis of predictive coding architectures [Rao and Ballard, 1999]. Forming predictions about future states supports high-level functions such as world model simulations since it allows the model to learn the transition structure in external environments. Within a modular architecture, predictive learning can be implemented for each module as a parallel generative module that predicts future inputs based on past states and updates the module's activation based on the prediction. During inference, prediction errors are minimized by running an optimization scheme on the inputs of the predictor module. Predictor modules inform the model of surprise and uncertainty through prediction error signals. In addition to their use as a learning signal for the predictor module, they assist the model in coordinating learning and credit assignment within the architecture. Examples of these predictor modules are sensory modules that predict incoming inputs and action modules that predict action outcomes. Other modules predict future rewards following actions and future states of the environment and the model. Predictor modules can be used during simulation conditioned on a world model structure. Visual predictor modules provide visual simulations of future states given action sequences. Prediction is not performed exclusively at short timescales; simulations of world models can progress at several levels of abstraction and at long timescales. Modules that are responsible for hierarchical planning can use their predictor counterparts to predict outcomes at their level of

abstraction and timescale.

4.4.2 Emergence of high-level functions

We believe that the modular architecture described in this section can perform a variety of high-level functions and computations, including meta-learning, compositionality, building world models, and hierarchical control.

Meta-learning Taking meta-learning as an example, it can be performed by a system that has access to information about task states, past rewards, and actions throughout many episodes of learning, as well as the capacity to manipulate its weights to find action policies that maximize rewards. The capacity of gated RNNs for meta-learning has been demonstrated in several studies [Duan et al., 2016, Kirsch and Schmidhuber, 2022, Hochreiter et al., 2001, Wang et al., 2017], and their weight manipulation is implemented with gating. The modular architecture described above could equate and potentially surpass gated-RNNs' meta-learning capacities since it implements various forms of gating, including top-down control, dynamic parametrization in certain modules, and routing, which can be assimilated to gating in implementations such as shared workspace [Goyal et al., 2022].

Building world models World model simulation requires representations of the model's variables and modules that learn the transition structure of the environment from experience. The representation of model variables can be distributed across relevant modules, and predictive modules learn the transition structure through the objective of predicting future states. The role of predictive modules in this case is to provide inputs for regular (i.e., bottom-up) modules when the model is in simulation mode. During inference, they are used to assert the correctness of the world model with respect to experience. The error signals they produce are used both for inference to signal unexpected outcomes and for learning to correct the world model. Taking maze navigation as an example, the model's controller can instantiate different algorithms for simulation and inference. Both algorithms involve model initialization with information such as the map structure, starting position, and goal position from sensory inputs to be maintained in short-term memory. Decision modules are used for selecting actions based on the current state, and prediction modules predict the future state and reward given the selected action. In the inference mode, the model receives the next state information

from sensory input, and prediction errors are saved for training. Whereas, in the simulation mode, the prediction itself is used as the next state representation and is used for selecting future actions.

Compositionality and structured representations Modular architectures are inherently compositional since they factorize computations and compose them according to task requirements. However, this architecture could additionally learn compositional representations using the inductive biases described above if trained under appropriate supervision. This entails the separation of concept representations from their multi-modal representations and their relations with other concepts. The associations of abstract representations to their multi-modal counterparts can be stored in model weights, such as the functions that translate representations between modules, or in memory modules. Inference over separate structure and content representations can be operated similarly to the TEM model [Whittington et al., 2020], where associations of representations of abstract roles within a structure and representations of sensory input are stored in memory online. Transitions within the structure, which are considered relational reasoning steps, can be performed by abstract modules dedicated to this function or predictive modules that learn to simulate these transitions from experience. The structures learned by the system can be generalized and flexibly modified using abstract representations that parameterize their instantiations in the relevant modules.

Hierarchical control To decompose tasks into their elementary components and build plans and atomic action policies, the representation of many levels of abstraction could be necessary. Task representations at higher levels of the hierarchy are maintained while the model executes lower-level task representations that compose them. The model could maintain hierarchical representations and representations of action sequences in memory modules or the hidden state of other recurrent modules. From the interactions of these modules with the main controller, a hierarchical control scheme could emerge whereby specific modules feed control policies to the main controller. Similar interactions could emerge for encoding many levels of abstraction in the representation of a structure.

These claims about the model's capacity to implement these complex functions lie on the assumption that it can learn them. Although I show examples of how these functions might be implemented, I believe that the model might not learn these functions even if it is provided with a carefully crafted curriculum and training scheme. These functions involve sophisticated algorithms that run

over several computation steps; they require proper execution with minimal error. In addition to the curriculum, which we describe in the following section, the model potentially requires additional inductive biases to promote the emergence of these functions. For example, the mechanisms of hierarchical control that could be useful for representing many levels of abstraction and executing hierarchical plans could be facilitated by a multi-layer gated RNN controller. To implement meta-learning, the model needs to maintain memory across training samples. This design principle could allow the model to leverage memories from past training samples to solve new ones.

4.5 Training and Curricula

The training process and experience are as important as architectural details for building an intelligent system. The complex architecture that we described requires several considerations for achieving successful training. Here, we will discuss the technical details required for training the controller-based modular architecture and the design of learning curricula.

Loss functions When considering inference in a modular architecture, we can observe that the learning problem involves two challenges: training the controller to select routing and module activation policies, and training the selected modules to process the routed information to solve the task. In an ideal situation, both learning problems could be trained with one objective, which consists of minimizing the loss function of the task. However, this is not possible when the task loss cannot be backpropagated through controller decisions. While routing decisions can be differentiable, hard routing schemes such as sampling from the Gumbel Softmax distribution [Maddison et al., 2017, Jang et al., 2017], known as reparameterization approaches, soft routing schemes using softmax or gating-based approaches such as the shared workspace, module activation, and environment interaction decisions are not differentiable. To train the controller on these decisions, prior work [Rosenbaum et al., 2017, Kirsch et al., 2018] used reinforcement learning with rewards designed based on task performance. The standard reward choice is the negative loss function. Following this objective, the controller is rewarded based on how low the loss function is driven by its decisions. Alternatively, an accuracy measure can be chosen as a reward function. In addition to the rewards obtained from the task, other rewards can be designed to guide controller decisions by penalizing actions that the module should not perform at specific

steps and rewarding actions that need to be performed. For example, the action of providing an output when no output is required from the task or reading inputs when none are available can be penalized, and reading inputs as soon as they are available can be rewarded. Other sophisticated examples include discouraging the model from activating modules without using their outputs. Furthermore, the number of computation steps can be constrained during early epochs using negative rewards to promote efficiency in the model. If the architecture is augmented with predictive modules, prediction errors can be aggregated to compute the loss used for training them. If the architecture is augmented with predictive modules, prediction errors can be aggregated to compute the loss used for training them. In total, the model learns using three signals: the task loss that trains modules to solve the task, the control loss that trains the controller and other modules involved in the control process, and the prediction loss that trains predictive modules.

$$\mathcal{L} = \mathcal{L}_{task} + \alpha\mathcal{L}_{actions} + \beta\mathcal{L}_{prediction} + \gamma\mathcal{R}$$

These losses are weighted with other module-specific regularizations. In a multi-task setting, losses from different tasks are modulated based on hyperparameters specified in the curriculum before they are used for the computation of the control losses $\mathcal{L}_{actions}$ of individual instances and added to the total loss.

Difficulties of training a modular architecture The subtleties of training modular architectures have been discussed in the literature. Rosenbaum et al. [2019] explains several factors of difficulty and failure cases often encountered when training modular architectures. Andreas et al. [2016a], Chen et al. [2020], Hudson and Manning [2019] also discuss the difficulty of learning visual reasoning programs in modular architectures, which can involve multiple training stages and sophisticated methods such as using a symbolic teacher [Chen et al., 2020]. The main source of these issues is the problem of learning meaningful activation and routing policies and functions executed by the modules simultaneously. The controller cannot learn how to use modules when they are randomly initialized, and the modules cannot learn if they receive inconsistent learning signals because of the random controller policies. This problem is a significant cause of training instability in early epochs. This problem is currently countered by using different learning rates for the controller and other modules or by using curriculum learning. Module collapse is another problem characterized by the controller policy converging to using a fixed routing path and training a specific set of modules. On the other hand, modular architectures can learn highly flexible policies that

overfit specific task instances and prevent generalization. These issues highlight the importance of balancing the flexibility of the controller policies. An additional problem that arises from the use of heterogeneous modules is an imbalance in their learning dynamics, which might result in the controller converging on suboptimal policies. For example, while one module learns fast, it could achieve lower performance than another module that requires training steps to reach its maximum accuracy. The controller policy could converge on using the first module without exploring the second module. Similarly, the lack of exploration can cause the controller to converge on suboptimal policies that involve many modules while discovering more efficient solutions. These issues are the reason for the difficulty of training stable modular architectures that generalize. They can exacerbate the difficulty of scaling modular architectures. The increased number of modules gives the controller a harder learning problem with a higher potential for overfitting simple tasks. Beyond the regularization techniques and various training schemes, the learning curriculum design can significantly mitigate these problems.

Curriculum learning Since the model learns action policies and computations from experience, its design and content are detrimental to the model's capacity for learning and generalization. For the model to learn diverse and flexible control sequences, it should be exposed to a variety of tasks during training. Additionally, for the model to implement a learning algorithm within its dynamics, the experience should challenge its capacity to learn novel rules from trial and error over several trials within one episode. Learning concepts from various modalities improves the model's abstraction. Furthermore, the model learns compositionality only if task performance or concept understanding requires the use of compositional representations. These are examples that motivate training the model in a multi-task setting that involves meta-learning tasks, tasks with multi-modal inputs, and compositionality as an inductive bias in the data. Compositionality can be introduced in the learning regime in several ways: tasks that require separate representations of structure and content, tasks that require hierarchical planning and inference, tasks that include atomic tasks, and compositional tasks presented in the learning experience in order from atomic to compositional. This idea is a special case of curriculum learning [Bengio et al., 2009] where experience is presented to the model in a gradual order of complexity. Curriculum learning has been shown to improve learning speed and generalization in ANNs. Although human experience is not based purely on a curriculum, the educational system organizes knowledge in a way that facilitates the learning of highly complex and

varied topics. Curricula can be beneficial for learning, but they are often difficult or impossible to create, especially for natural data, which is arguably the most important domain for training AI systems. Fortunately, the AI field is witnessing the emergence of many tasks that can be generated based on programs—procedurally generated environments, for example. Furthermore, there are methods for estimating the difficulty of task instances based on the performance of previously trained models. Although biased by the reference model's architecture, these difficulty estimates can be used to create an artificial curriculum for the task. Nevertheless, while the learning system might greatly benefit from curricula during initial training, once it acquires the skill of decomposing tasks into their atomic components, it could generalize this capacity to tasks that are not organized as curricula.

Curriculum design The first point to consider when designing a curriculum is the skills that we want the model to learn. The tasks that compose the curriculum are chosen so that they involve the use of targeted skills. For the model to learn how to control spatial or feature-based attention, it must learn a task that requires the use of attention mechanisms. Thus, a task such as object recognition from images with centered and fixed-scale objects would not be suitable for learning this skill. An important point to consider is that the design of the task should ensure that models cannot exploit shortcuts to solve the task without employing the target skill [Geirhos et al., 2020a]. Given a set of skills and a set of tasks, the curriculum can be created by decomposing each task into its elementary components and then parsing factors of variation in each task that control the task's difficulty and complexity. These steps are manageable in a synthetic dataset, such as toy tasks (repeat copy [Graves et al., 2014], maze navigation, among others), cognitive tasks [Yang et al., 2019], abstract reasoning tasks (CVR, Raven [Zhang et al., 2019], among others), mathematics (arithmetics and general math tasks [Hosseini et al., 2014, Mishra et al., 2023], among others), simulated RL environments and games (avalon [Albrecht et al., 2022], among others). The task parameters could control different dimensions of difficulty for different skills. Given assumptions over the complexity of these skills, the parameters can be used to constrain the number of inner processing steps according to the complexity of the skill. If we take the repeat copy task [Graves et al., 2014] as an example, it is a task that requires memory manipulation skills primarily; it also requires the controller to learn how to run nested loops over lists, which involves counting and producing input patterns from activations (auto-encoding). The main parameters of this task are the length of the input list and the number of repeats. Other parameters include the distribution of input vectors; these parameters can be used to test the

out-of-distribution generalization capacities of the model. The parameters of this task decompose into two component tasks: the copy task when the repeat parameter is 1, and the repeat task when the list length is 1. The latter requires mainly looping skills, while the former requires both memory and looping skills. While it is possible to decompose a synthetic task and control its parameters, this is not the case for tasks based on recorded data, such as natural image datasets. This is among the reasons that restrict the adoption of curriculum learning techniques in AI research. Although it is not possible to create a rigorous curriculum over a wide range of tasks and benchmarks, several techniques can be used for approximating the difficulty of individual task instances, allowing for the creation of a curriculum. Among these techniques are heuristics on the data, such as text length and the number of uncommon words in NLP tasks, and the accuracy and prediction confidence of previously trained models in specific instances. Alternatively, difficulty can be introduced into the data using additional transformations; image augmentations such as cropping and distortion are examples.

Scheduling tasks and difficulty The curriculum is defined by the tasks, their scheduling, and the scheduling of their parameters. The standard scheduling choice is to progress training from simple to complex tasks and from low to high difficulty in each task. Tasks are preceded by tasks that compose them in the curriculum; the repeat copy task would be preceded by the repeat task and copy task. An important choice in the curriculum design is progress triggers. Progress in the curriculum can be fixed based on the number of training steps. A more intuitive method is progressing in difficulty based on model accuracy, as in automatic curriculum learning [Portelas et al., 2020]. For example, difficulty can be increased when the model reaches a pre-specified level of training validation accuracy and decreased when the model performance degrades. To avoid catastrophic forgetting, easier levels of difficulty can be randomly sampled throughout training.

Evaluation Following training, the model can be evaluated with respect to the intelligence factors discussed above. As in the standard evaluation process in machine learning, performance can be measured by the model's accuracy on held-out in-distribution and generalization test sets. Energy and time efficiency can be measured with the training speed (the number of training steps required by the model to learn a task) and the computational efficiency (the number of inner recurrent steps required by the model to reach a solution and the total number of module activations). Data efficiency can be measured by the number of unique task instances required by the model to learn a task. It can be measured over many train-

ing runs with different numbers of training samples. These factors are measured mainly with respect to performance; it is necessary for the model to be capable of solving the task for measures other than performance to be taken into consideration. If the model were trained in a standard setting without a curriculum, these measures would characterize only the model. However, in a curriculum setup, the evaluations also reflect the quality of the curriculum. More specific generalization capabilities, such as compositional generalization and compositional learning, can be evaluated using curriculum and reverse curriculum settings similar to the ones developed in the CVR benchmark.

Adapting tasks to the framework The multi-task setting with heterogeneous tasks requires technical considerations regarding the model and task components. Heterogeneous tasks involve the processing of various types of data as single instances, sequences, and lists following a specific temporal organization. For example, a visual reasoning task such as Raven or CVR provides a list of images as inputs and expects a categorical decision as a symbolic output; a VQA task such as CLEVR provides one image and a list of text tokens as inputs and expects one text token as an output. The repeat copy task gives a sequence of n-dimensional vectors as inputs and expects a sequence of similar vectors as outputs, but the sequence of outputs is expected only after the model has read the sequence of inputs. For one model to be trained on these tasks simultaneously, it must be equipped with input and output modules for processing the tasks' inputs and outputs. Similarly, the task instances must be defined such that their inputs and outputs are mapped consistently to specific modules. To organize this process, the data is categorized into types: symbolic, image, and text, and streams are categorized into single, list, and sequence. This task-specific information is used for selecting the model's input-output modules and mapping task inputs and outputs to model external gates. When a task sample is generated, each input and output is associated with a timestamp that specifies when they can be read or expected from the model. The timestamp is defined as several steps. As the model's controller makes decisions about task interactions, the task sample provides inputs and outputs based on its step counter. Taking the repeat copy task as an example, the inputs are a sequence of vectors and the number of repeats; the data type is symbolic for both; the stream type is sequential for the vectors and single for the number. The task requires one output, which is a sequence of vectors, which are symbolic data, and a sequential stream.

4.6 The abstraction network

To provide a proof of concept for the framework described in previous sections, we developed AbstractNet, a neural network that incorporates modularity with diverse modules, routing, and adaptive computation time. Given the capacity to control its internal computations and interactions with tasks, AbstractNet can leverage the expressivity of its modules and routing schemes to implement diverse algorithms and solve many tasks. The model's architecture is composed of several modules, including the *controller* module. Each *module* is defined by its architecture, the *actions* that it can perform, and the *gates* through which it receives and sends information. AbstractNet interacts with the tasks using two additional actions: *"read input"* which reads input from the task, and *"update output"*, which provides output to the task. Actions of one module can use different gates; for example, a memory module has a *read* and *write* action and gates for querying memory *"query"*, adding information *"value"*, and memory output *"output"*. The *read* action uses gates *"query"* and *"output"* while the write action uses *"value"*. Gates are differentiated based on their use in the model: external gates interact with task variables by processing inputs or providing outputs; internal gates transfer information between modules; and gates that receive and send information to only one module process recurrent states of the module. The architecture is explained in Figure 4.3.

The main modules that compose AbstractNet are the controller and the input and output modules chosen based on task specifications. Other modules include memory modules and general computation modules that have exclusively internal gates. The controller is a special module because its outputs determine action decisions and the routing of information between modules. The controller architecture is a gated-rnn GRU [Cho et al., 2014] supplemented with MLPs for encoding the internal state of the system (routing matrix, action decisions, and task state), value prediction, actions, routing decisions, and inputs from other modules through an internal gate. Input and output module architectures depend on the task. We use a CNN for visual inputs and MLPs for standard N-dimensional vector inputs and outputs. These modules are used for single-unit and sequential data. List inputs are processed using list modules, which include a positional embedding and select input elements based on queries using softmax-attention. Among a variety of memory modules, we experiment with the memory system used in DNC [Graves et al., 2016]. Additional module architectures include MLPs, gated

RNNs, and hypernetworks with varying numbers of input and output gates and layers. These modules are additional computational resources selectively used by the model for learning computations that are generalized across tasks and task samples.

The model routes information between modules using translator modules that transform representations given vector embeddings of the source and the target modules. The routing matrix predicted by the controller is used for aggregating translator outputs. We also experiment with simpler routing schemes that do not use a translator, although this forces modules to share the same representational space, which could limit the architecture.

During inference, the model maintains many representations of its recurrent state, including the recurrent states and outputs of all modules. These representations are initialized according to the specifications of each module, used as inputs, and updated throughout inference iterations. At each iteration, the controller outputs one probability value for each action, which is used for sampling a decision from the Bernoulli distribution. It also outputs a routing matrix, which has internal input gates as rows and internal output gates as columns. Each line of the matrix is normalized with Softmax and used as weights for aggregating translator outputs. The model stops inference iterations when the task state signals the end of execution. The model can perform several iterations without interacting with task variables. The number of these iterations is limited to ensure that the model finishes solving task samples. The inference process is displayed in Figure 4.3 and Algorithm 3.

AbstractNet is trained end-to-end with task-specific objectives. The weights of modules and networks used for routing are optimized using task-specific losses, while networks used for deciding module activation and task interactions are trained in a reinforcement learning setting. We use advantage actor-critic (A2C) with generalized advantage estimation (GAE).

$$\mathcal{L} = \mathcal{L}_{task} + \alpha \mathcal{L}_{actions}$$

where α is a hyperparameter that weighs the two losses. The reward is chosen as the negative loss computed for the task sample $r = -\mathcal{L}_{task}$. Models with an ablation of the adaptive computation time do not include module activation and task interaction decisions; they are trained with the task-specific loss only.

Algorithm 3: Inference process

Data: Task sample d
Initializations:
$m_{state} \leftarrow 0$ `// model state`
$h \leftarrow 0$ `// model output representations`
$c_{in} \leftarrow 0$ `// controller input`
$o_{ext} \leftarrow []$ `// output buffer`
while $d_{state} \neq DONE$ **do**
  $M, a_p, l_p, c_{out} \leftarrow controller(c_{in}, m_{state}, d_{state})$
  $a \leftarrow sample\_actions(a_p)$
  $x_{in} \leftarrow setup\_inputs(h, M)$
  $h' \leftarrow run\_actions(a, h, x_{in})$
  $o \leftarrow update\_output\_buffer(o, h', l_p, a_p, a)$
  $h \leftarrow update\_state(h, h', c_{out}, M, a)$
end

4.7 Experiments

In this section, we present initial experiments and preliminary results on the AbstractNet architecture. These experiments aim to evaluate the model's capacity for learning and composing routing schemes for various tasks and adapting computation time based on task requirements. With this goal, the model is trained on many tasks with a variety of computational requirements in single- and multi-task settings. The experimental setting in this study is restricted to simple tasks and a version of AbstractNet that incorporates modularity, soft routing, and adaptive computation. This framework allows for validating and analyzing the function of the architecture on a small scale before tackling the challenges of training a larger version of the model on more complex tasks.

4.7.1 Tasks

We first examine the model's capacity for learning various tasks in an end-to-end fashion. The model is trained on several tasks with various computational requirements.

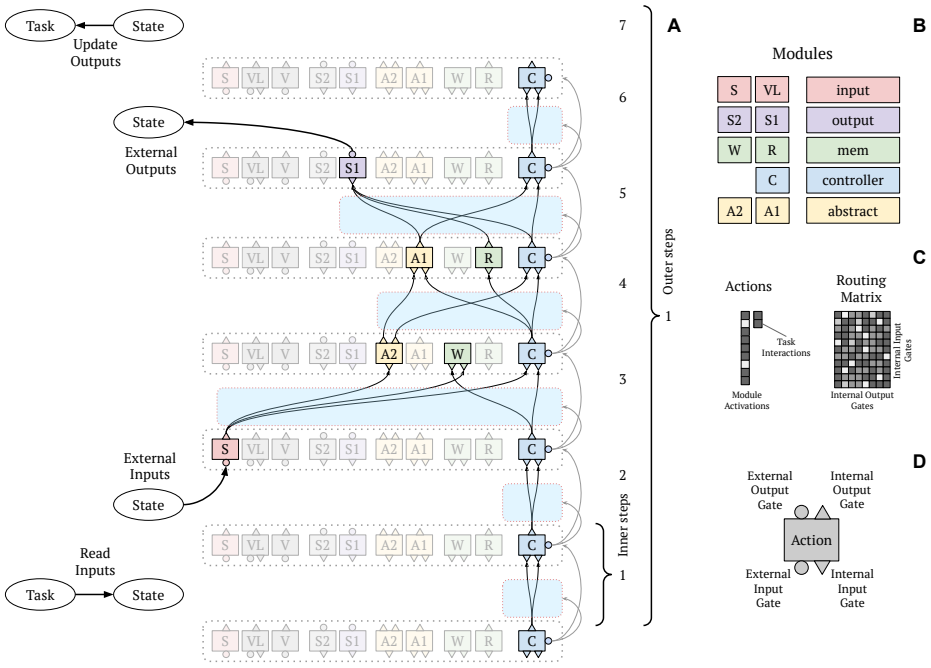


Figure 4.3: **AbstractNet Architecture** a) In model inference, the controller selects actions and routing matrices at each processing step. Decisions to read inputs and update outputs are independent of module activity. The model remains in the inner computation loop until it updates the output. The outer steps follow the task's progress. b) The list of modules represented in the inference process. c) The controller determines routing matrices, action decisions, and task interactions. c) The difference between external and internal input and output gates.

Visual categorization Small image datasets MNIST [Deng, 2012] and Cifar10 [Krizhevsky et al., 2014] are used for evaluating the model's capacity to learn simple routing schemes. In this task, the model is equipped with a vision module that is used for processing images from both datasets and two classification modules, one for classifying MNIST images and the other for Cifar10 images. The loss computed in this task is a cross-entropy loss over the logits provided by the classification modules. The model can solve this task by routing the output of the vision module to the correct classification module.

Selection task A synthetic list manipulation task in which the model selects an element from a list based on its index. The model is fed a list of 16-dimensional vectors that range in length from 2 to 20 elements and the index of the element to

retrieve as a scalar. The model is equipped with a list manipulation module that uses a differentiable attention mechanism to read and encode elements from the list. The list manipulation module selects elements by processing the query vector, performing a dot product with positional embeddings of the elements, and then weighting the element vectors by a softmax transformation of the dot product. The output of this task is a 16-dimensional vector, which is passed through a sigmoid activation function and used for computing the L2 loss. This task involves two input streams and can be solved in a two-step fashion: reading the index and using it to select the element from the list. The difficulties of this task lie in learning the correct mapping from index to positional embedding and simultaneously learning to encode and decode the element vectors correctly.

Cognitive tasks The model is trained on a set of cognitive tasks that were developed by Yang et al. [2019]. These tasks are used for investigating the cognitive capacities of humans and animals, such as attention and working memory, and the learning of abstract rules. The general task design consists of a sequence of N-dimensional vector inputs that represent the display of a fixation and dots on a screen. Each bit of the vector corresponds to either the fixation or one of the dots that form one of two rings representing two modalities. The model outputs a vector at each timestep that represents its choice and is used to compute the L2 loss with the target vector. The model is trained to choose specific stimuli based on the task design. The tasks can be grouped into different families as described in Yang et al. [2019]; the Go tasks, in which the response is expected in the direction of the stimulus, Anti-tasks, where the response should be opposite to the stimulus, DM tasks where two stimuli are presented and the response is associated with the stronger stimulus, Dly DM tasks, which are similar to DM tasks with a temporal separation between stimuli, and matching tasks, where two stimuli are presented and the response depends on whether they match, The tasks of each family differ in details that pertain to fixation, order, time, and modalities but follow the same logic. For example, in the fdgo task of the Go family, the model is trained to choose the direction of the fixation mark until it disappears, after which it chooses the direction of the stimuli. Samples in these tasks are generated randomly by varying the modality of the stimuli, their direction, and the time delays between phases of the task. These tasks are interesting since they evaluate the model's capacity for learning abstract rules and flexibly changing routing schemes based on task demands.

Copy task The copy and the repeat-copy tasks evaluate the use of memory.

They were used for testing NTM [Graves et al., 2014] and DNC [Graves et al., 2016]. The model is fed a sequence of N-dimensional random vectors as inputs. After reading all elements of the sequence, the model outputs a sequence of similar length. The model is trained to output the same sequence of input vectors. The L2 loss between the output and target sequence is used for training the model. The length of the sequences ranges from 2 to 20 elements. In the repeat copy task, the model is trained to repeat the same sequence over many iterations. The number of iterations is input as a scalar. These sequential tasks involve memory manipulation and changes in routing schemes for different phases of the tasks: reading inputs and storing them in memory or reading memory and outputting vectors. The model is equipped with a differentiable memory module, as in DNC [Graves et al., 2016], to solve this task.

bAbi task The bAbi question answering dataset [Weston et al., 2015] evaluates language understanding through reasoning over linguistic facts presented in a story. It consists of 20 synthetic tasks that test different types of reasoning over language, including inference, counting, time reasoning, positional and size reasoning, and path-finding. Each sample consists of a story composed of many sentences and a question that is answered with a single expression. The model uses story elements as supporting facts for answering the question. Sentences are embedded by applying a positional encoding to each word and summing all embeddings; we chose this encoding following Henaff et al. [2017], Dehghani et al. [2019]. Sentence embeddings are fed sequentially to the model, with the question as the last input. Models can be trained on 1000 samples or 10,000 following the benchmark. The experiments in this chapter focus on the 1000 and 2000 sample data regimes.

Each task was prepared for training by specifying the inputs, outputs, and timestamps, as well as the minimum and maximum numbers of internal computation steps allowed for the model within each task step. The model architecture is built based on each task's input and output gates. Model and task specifications include the mapping between task variables and module gates. For example, the model built for image classification, as in Figure 4.6, includes a vision module and two classification modules for the two image datasets (MNIST and Cifar10), while the model built for the selection task uses a list manipulation module to read and query the list, another input module to read the index, and an output module.

4.7.2 Baselines

In single-task training, AbstractNet is compared to task-specific architectures. A CNN with a classification head is trained on image classification since CNNs are the standard models for image processing; a GRU is trained on cognitive tasks similarly to the models presented in Yang et al. [2019]; and a list attention-based model is trained on the selection task. In the list attention-based model, the index is embedded and fed as a query to the list attention module, and the output is decoded and used for computing the loss. Finally, a memory control model, Differentiable Neural Computer (DNC) [Graves et al., 2016], is trained on the copy task.

In the multi-task setting, transformer models that use the self-attention mechanism [Vaswani et al., 2017] are interesting as a baseline since they demonstrate impressive multi-tasking capacities at large scales. The simple transformer encoder is a monolithic architecture composed of layers of MLP transformations and multi-head self-attention with skip connections. Taking several tokens as input, these operations can be thought of as refinements of input representations based on other tokens. The self-attention mechanism can be thought of as a constrained form of information routing between token representations. I selected Universal Transformer [Dehghani et al., 2019] as a baseline since it is comparable to AbstractNet in terms of adaptive computation time. The implementation of Universal Transformer (UT) is adapted to our framework for a fair comparison with AbstractNet. My implementation of UT is based on a pytorch implementation, which is an adaptation of the original architecture implemented in this repository. UT is augmented with the same external input and output modules that are provided to AbstractNet for different tasks. In contrast to recurrent architectures, transformers process sequential inputs in parallel as a list of embeddings masked with a positional embedding to mark their position within the sequence. To differentiate inputs from their sources, we use a similar masking technique where each input is masked by its gate embedding. Additionally, UT is probed for outputs by providing an additional input token that corresponds to the output gate embedding. The model refines the representation of these embeddings using information from other tokens through self-attention to build the corresponding output vector representation. In sequential tasks, such as cognitive tasks, UT is provided a history of input embeddings up to an input size limit, considered a history of inputs, since it does not have access to memory modules and does not have a recurrent state like RNNs.

Input embeddings are masked with input gate embeddings for each module. The required outputs at each processing step are specified, with output gate embeddings as additional inputs. Since transformer-based models do not have memory, in sequential tasks, the model is provided inputs of previous and recurrent states as history.

In the basic version of the model, all modules are active if they have available inputs, and decisions about task interactions (reading inputs and providing outputs) are fixed based on the minimum and maximum number of internal steps specified by the task. When the model is trained with adaptive computation, it can select actions based on task interactions, which allows it to adapt the number of computation steps to the task demands. Available actions are sampled from a Bernoulli distribution using model outputs as probabilities. The model is encouraged to find efficient solutions using rewards for task interactions that decrease linearly with the number of internal computation steps.

All models are trained using a fixed set of hyperparameters for all tasks. For AbstractNet, the Adam optimizer is used for updating model weights with a learning rate of 0.0002, no weight decay, and a batch size of 30. The dimension of the input and output vectors of abstract modules is 128. The routing process does not involve a translation module. The controller is a GRU with a hidden vector of size 512.

Single task training results In the first experiment, we train the model on each task individually. Results in Table 4.1 show that the model can learn to perform several tasks without requiring many training samples. These tasks include ones that require simple input-output mapping schemes, such as the selection task and visual tasks, and others that are sequential and require adaptive behavior, such as cognitive tasks and the sequence copy task. These results highlight the model's capacity for learning routing schemes, allocating computation time, and solving tasks using modules using one optimization method over all model weights, in contrast to other methods that separate controller training from module training. Interestingly, the model is capable of manipulating modules that incorporate a variety of inductive biases, including MLPs, a CNN, a memory module, and an attention-based module. The limited performance of the model on visual tasks can be explained by the shared visual module for classifying MNIST digits and Cifar10 objects and by the limited number of training samples. Accordingly, increasing the number of training samples from 2000 to 60000 improves the performance on these tasks, as shown in Table 4.1.

Task	Model	N samples	N steps	Accuracy
Visual	AbstractNet AC	2000	6.15	64.79%
	AbstractNet	2000	8.00	65.95%
	Universal-T ACT	2000	-	66.17%
	Universal-T	2000	-	66.51%
	CNN	2000	-	66.67%
	AbstractNet AC	60000	5.47	71.28%
	AbstractNet	60000	8.00	78.65%
	CNN	60000	-	79.40%
Selection	AbstractNet AC	2000	6.74	87.62%
	AbstractNet	2000	8.00	85.68%
	Universal-T ACT	2000	-	49.99%
	Universal-T	2000	-	50.00%
	List-Attention	2000	-	70.74%
Cognitive	AbstractNet AC	2000	3.08	92.26%
	AbstractNet	2000	4.79	95.99%
	Transformer-T ACT	2000	-	97.91%
	Transformer-T	2000	-	97.98%
	GRU	2000	-	97.27%
Copy	AbstractNet AC	2000	2.16	90.35%
	AbstractNet	2000	4.00	99.99%
	Transformer-T ACT	2000	-	80.61%
	Transformer-T	2000	-	99.99%
	DNC	2000	-	99.99%
Babi	AbstractNet AC	1000	1.62	70.79%
	AbstractNet	1000	3.55	75.05%
	Universal-T ACT	1000	-	60.68%
	Universal-T	1000	-	60.99%
	Universal-T ACT - paper	1000	-	95.45%
	Universal-T - paper	1000	-	94.69%

Table 4.1: **Performance on individual tasks**: The performance of AbstractNet and UT often approaches that of task-specific models. AbstractNet AC, which is trained with adaptive computation (AC), uses systematically fewer computation steps compared to AbstractNet, which uses the maximum number of computation steps specified by the task. The AC-trained models find efficient solutions to all tasks.

As shown in Table 4.1, AbstractNet AC (adaptive computation) reaches a similar performance to AbstractNet on most tasks, which shows that training with adaptive computation does not significantly deteriorate performance. Furthermore, models with adaptive computation use fewer internal computations during inference. For example, the model reduces the number of internal computations from 8 to an average of 5.47 in visual tasks and from 4 steps to an average of 2.16 in the copy task. The numbers of internal steps are determined by stochastic action decisions; they are averaged over outer steps for each sample and test set samples of each task. Considering the image classification task, if the model's strategy relies only on the visual and decision modules, the most efficient solution would be routing the visual module output to the decision module input. The resulting number of computation steps would be 4, which corresponds to: 1) receiving visual inputs from the task; 2) encoding the input using the vision module; 3) processing the visual embedding with the decision module; and 4) routing the output to the task. When analyzing inference episodes, we observe that the model follows this strategy with many task samples, as in Figure 4.6, where visual embeddings are routed to model decision modules and other abstract modules are mostly unused. Figure 4.4 shows the progress of the number of internal computations over training. The model initially increases the number of inner computation steps to the maximum specified by the task. The number of computation steps decreases only after accuracy reaches its peak. This progress can be sensitive to the magnitude of the reward used to encourage the model to decrease the number of computation steps. These results are indicative of the model's capacity to adapt the number of internal computation steps and find efficient computation sequences while solving tasks.

Results in Table 4.1 also show that AbstractNet is competitive with the proposed baselines in this restricted setting and single-task training as it's capable of performing on par with task-specific baselines and UT. While AbstractNet surpasses the performance of UT on the selection and copy tasks, it does not perform better in other tasks. The differences between the accuracy of UT on bAbi and the accuracy reported in Dehghani et al. [2019] is due to the fixed training settings and hyperparameters for the two models. These results are achieved without hyperparameter tuning for all models, including UT. This explains the difference in performance of UT on the bAbi tasks between the results reported in Dehghani et al. [2019] and the experiments of this study.

Curriculum learning Among the main challenges of training AbstractNet are

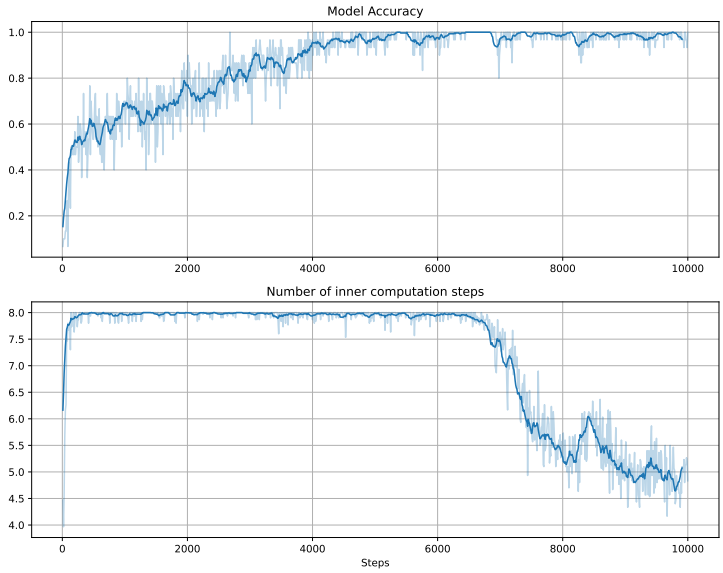


Figure 4.4: **Adaptive computation time**: training accuracy and the average number of internal computation steps are shown across training steps on image classification. The number of internal computations increases to a maximum of 8 steps early during training and decreases after accuracy increases. The model finds more efficient solutions after learning the task.

the joint training of individual modules, routing in a modular architecture, and task interaction decisions. The random initialization of routing schemes and module weights causes unstable learning in early epochs, resulting in suboptimal performance or divergence. These issues are more prominent when learning complex tasks. A solution to this issue is the use of curriculum learning. The synthetic tasks can be organized in a curriculum where the levels of difficulty are specified by the length of the input sequence. In a second experiment, we train AbstractNet on the copy task following a curriculum that increases difficulty after the model has surpassed a preset threshold of validation accuracy. Figure 4.5 shows the difference in training progress between a model trained following a curriculum and another model trained using randomly sampled task instances. The curriculum-trained model reaches maximum accuracy on the hardest difficulty level before the standard model converges. This result hints that curriculum learning improves the learning dynamics of our modular architecture.

Multi-task training results To evaluate the model for multi-task learning, we

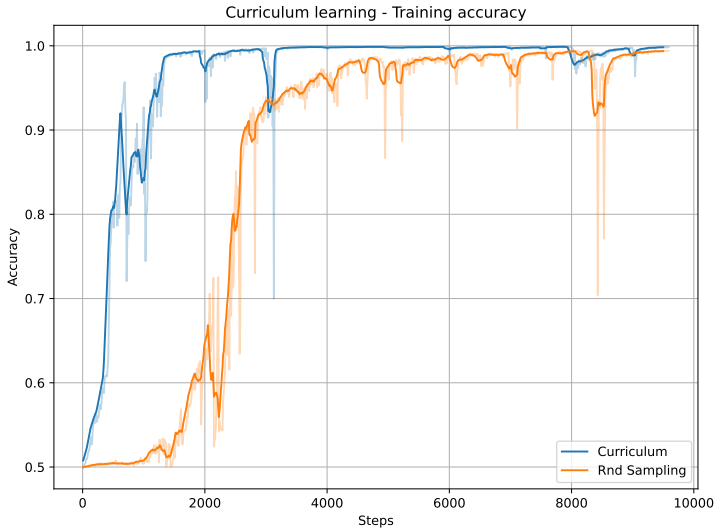


Figure 4.5: **Curriculum learning**: training progress of two models trained on the copy task. The model trained using a curriculum reaches maximum performance in fewer training steps than a model trained on randomly sampled task instances.

devised two experimental settings comprised of homogeneous tasks and heterogeneous tasks. Homogeneous tasks share modules and routing strategies; they test the model's capacity to reuse modules in various contexts and learn versatile routing strategies for different tasks. Heterogeneous tasks, on the other hand, require the use of different modules to test the model's capacity for manipulating many modules and learning many independent routing strategies. In these settings, batches are uniformly sampled from all tasks, and losses are aggregated from different task instances without using weights. Tables 4.3 and 4.2 summarize the results of these experiments. The tasks chosen for the homogeneous task setting are the cognitive tasks and the bAbi dataset since they are all sequential tasks with similar inputs and outputs. AbstractNet successfully learns cognitive tasks both with and without adaptive computation and reaches a decent performance on bAbi tasks. Although it does not surpass the accuracy of UT in most settings, it remains on par with its performance. In the heterogeneous task setting, the model is trained on "fdgo," one of the cognitive tasks, the first bAbi task that involves inference from one supporting fact, and other tasks that the model solved in the single task setting. AbstractNet is competitive with UT when trained on all five tasks. Furthermore, when it is trained on four tasks, excluding the bAbi tasks, it is

Model	Cognitive	bAbi
AbstractNet AC	93.07%	70.99%
AbstractNet	92.56%	76.65%
Universal-T ACT	95.11%	62.47%
Universal-T	95.81%	80.30%
Universal-T ACT - paper	-	92.22%
Universal-T - paper	-	91.50%

Table 4.2: **Multi-task performance on homogeneous tasks**: The cognitive task set and bAbi dataset contain both 20 homogeneous tasks. 2000 training samples are used for each task. AbstractNet reaches a higher accuracy on cognitive tasks compared to the single task setting and a similar accuracy on bAbi tasks.

capable of solving most of them. Although the model learns to solve these tasks, it has lower performance compared to the single task setting, especially on the copy task when trained with AC. These results demonstrate the model's capacity to multi-task various types of tasks and its flexibility.

Model analysis To understand how AbstractNet solves tasks, we visualize the routing matrices during inference in Figures 4.6 and 4.7. Figure 4.6 shows inference on image classification and selection tasks; the models are trained on these tasks individually with adaptive computation. In the image classification task, the controller routes the visual input to all modules, including the output decision module. It decides to read task inputs in the first step and emits the output to the task as soon as the decision module receives the visual representation. In the selection task, after reading inputs from the task, the model first encodes the index and routes it to the list module as a query. The output of the list module is routed to the decoder module, and then the output is emitted. In both tasks, the model learns the minimum number of steps necessary for solving the task. Furthermore, it only uses one routing scheme across all task samples because these tasks require only simple interactions between modules that can be performed without interference using one routing scheme.

Analysis of sequential tasks such as fdgo reveals interesting inference dynamics. In the fdgo task, the model is presented with input that contains a fixation mark and stimuli at specific locations. The task is to output the location of the fixation mark until it disappears, and then to output the location of the stimulus. Each task sample consists of three distinct phases: the presentation of the fixation mark,

Model	fdgo	Visual	Selection	Copy	bAbi 1
AbstractNet AC	98.30%	66.69%	76.94%	68.33%	99.60%
AbstractNet	97.79%	65.56%	77.16%	56.47%	99.30%
Universal-T ACT	98.77%	65.34%	64.74%	82.38%	67.70%
Universal-T	94.52%	64.81%	49.99%	60.52%	99.80%
AbstractNet AC	98.36%	67.14%	96.17%	75.62%	-
AbstractNet	97.85%	65.70%	94.64%	95.73%	-

Table 4.3: **Multi-task performance on heterogeneous tasks**: The heterogeneous task set contains one cognitive task ("fdgo"), question-answering tasks (questions that require one supporting fact), and other tasks the model can solve in the single task setting. Models in the third row were trained on all tasks except the bAbi task. These results show that AbstractNet is capable of reaching single-task performance on most tasks while learning them in a multi-task fashion.

the introduction of stimuli, and the disappearance of the fixation. We observe that the controller routing matrices change across these phases, as shown in Figure 4.7(b). Visualizations of the routing matrices at these three phases show that there are many similarities: the controller receives signals predominantly from the input encoder, which allows it to adapt the routing scheme based on changes; the output module receives input predominantly from the abstract module 5; and several abstract modules process the task embedding. The changes between the three phases are mostly changes in routing between abstract modules. An important change is the input to the abstract module 5, which mostly decides the output. Consistently with the task demands, the module's inputs do not change as long as the model must focus on the fixation; they only change in the third phase, where it receives inputs from abstract module 2 and the task embedding. The changes in inference dynamics can be observed in other sequential tasks, including the copy task.

4.8 Discussion

In this chapter, we have presented a model and a general framework for designing neural networks inspired by aspects of brain function. Our work relies on biological priors from the brain and learning experience to build this framework. The biological priors of the brain, such as memory, attention, control, and pre-

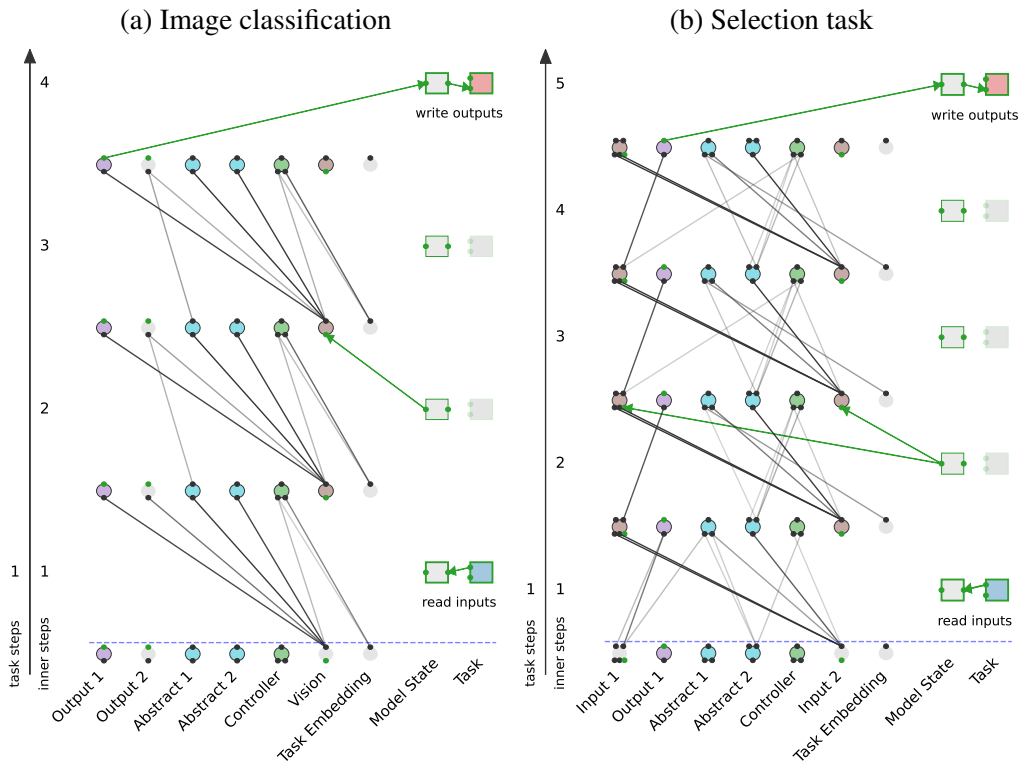


Figure 4.6: **Routing matrices**: Visualization of inference on the image classification and selection tasks. Inactive modules are represented as gray circles, and task inputs and outputs are represented as green lines. AbstractNet finds the minimum number of computation steps to solve these tasks and uses an overall one-routing scheme over all computation steps.

dictive learning, have provided AI researchers with many inspirations for general and specialized neural network architectures. These biological prerequisites alone are not sufficient for building an intelligent system; the learning experience plays an important role in shaping the system. The structure of the human experience is characterized by compositionality and its curricular nature. This framework unites modularity, adaptive computation, recurrence, a variety of inductive biases, and curriculum learning to build a neural network architecture and its learning experience. To provide an example and a proof of concept of these ideas, we propose AbstractNet, a modular neural network architecture, and through initial experiments, we show the model's capacity to learn routing schemes, internal

computation decisions, and distinct module functions in an end-to-end fashion. The model can multi-task homogeneous and heterogeneous tasks, adapt its computation time, and flexibly manipulate modules to perform various functions.

Although our presented experiments encompass various aspects of the model, they only provide a partial exploration of the overarching framework. The employed version of AbstractNet in these experiments focuses solely on modularity, module routing, and task interaction decisions, lacking module activation decisions, top-down modulation, predictive modules, and control over the learning signals. Moreover, the diversity of modules remains limited, as we omit the investigation of hypernets, RNNs, and attention-based abstract modules. Curriculum learning experiments are preliminary since they show only that AbstractNet benefits from curriculum training. AbstractNet is not the only network that benefits from curriculum training since its benefits have been demonstrated in a host of models and on a variety of tasks (read Soviany et al. [2022] for a survey). More interesting investigations on the impact of curriculum learning can compare the learning speed and generalization of different models trained with similar curricula.

Furthermore, our experimental evaluation neglects crucial aspects such as meta-learning, compositionality, and out-of-distribution generalization. Additionally, the experiments are constrained to toy tasks, restricting the investigation to small-scale scenarios. Notably, we observe that the current model faces challenges in solving more intricate tasks, including meta-reinforcement learning (meta-RL) tasks, due to training instability and limited exploration post-convergence. We believe that these learning issues could be alleviated through the use of tailored curricula and regularization techniques, and we defer these concepts for future advancements.

Our framework introduces fundamental principles for designing, implementing, and training models. A central assumption is that a single model, equipped with diverse inductive biases, has the potential to embody an efficient system. While our model demonstrates promise for addressing compositionality and meta-learning, it may not excel in terms of inference efficiency. Inference involving controller intervention at each processing step can lead to slower execution. In contrast, the brain adopts a strategy of fast learning and slow inference, yet it could potentially leverage a strategy of slow learning for rapid inference, resembling cortical learning in the CLS framework. AbstractNet currently possesses the capability to simulate such a learning strategy within the weights of individual

modules. However, due to its limited ability to add connections between modules beyond routing, it cannot efficiently perform controller-free operations for complex tasks involving more than one module.

The framework assumes the feasibility of constructing curricula for any given task. However, this assumption hinges on the ability to decompose each task into its constituent subtasks, which may not hold universally. Even in cases where decomposition is possible, it necessitates substantial manual effort for design. Moreover, manually designed curricula incorporate the researcher's priors, potentially introducing biases to the model's learning process and restricting its capacity to explore alternative task decompositions. Additionally, the issue of benchmarking models trained on intricate curricula remains challenging and has not been adequately addressed within the framework. Benchmarking in this context should consider both the model's progress within the curriculum and the amount of experience employed.

This work primarily serves as a conceptual exploration, aiming to inspire a fresh methodology in the development of brain-inspired neural network models towards achieving human-level intelligence. The presented experiments serve as a prototype for the proposed model, providing a proof of concept regarding its underlying mechanisms. Our ongoing research delves into various avenues of further development within this framework, to stimulate and motivating fellow researchers to expand upon these ideas.

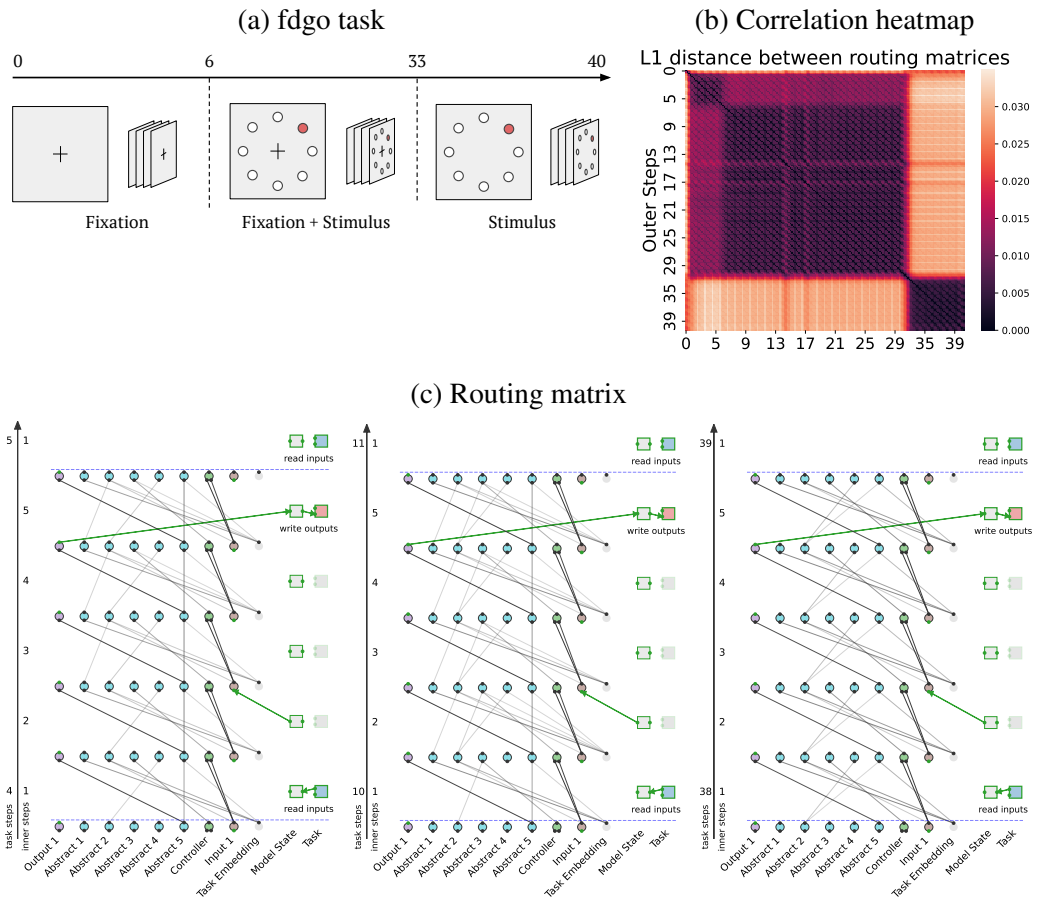


Figure 4.7: **Routing analysis**: AbstractNet adapts the routing scheme based on task demands. a) The fdgo task consists of three phases: a fixation phase, a fixation+stimulus phase where the model must return the fixation location and a stimulus-only phase where the model must return the stimulus location. b) The heatmap represents the L1 distance between the routing matrices of two successive sets of six computation steps. The three phases of the task can be distinguished by periods of similar routing matrices: 0-6, 6-33, and 33-40. c) Inference visualizations from the three phases show overall similarities in the routing of the embedded input to the controller and output from abstract module 5. The difference is significant in the routing between abstract modules.

CHAPTER 5

DISCUSSION AND FUTURE WORK

Within the large field of AI, this dissertation addresses an important characteristic of cognitive function that deep learning models fail to incorporate to the extent reached by human brains: the principle of compositionality. Compositionality is hypothesized as a tool for organizing thought, which is demonstrated in several facets of cognitive function, such as representing knowledge and learning tasks. Several studies probe compositionality in deep learning models and attempt to incorporate compositional computation into them. Nevertheless, certain aspects of compositionality, such as skill composition and decomposition during learning, are yet to be explored in deep learning. I focus on these aspects of learning and relate them to sample efficiency—the amount of experience needed for learning a novel task. I developed a benchmark for compositional visual reasoning (CVR) that evaluates compositional learning and sample efficiency in humans and neural networks. The experiments have demonstrated a large gap between the sample efficiency of humans and deep learning models that are pre-trained on a variety of visual tasks. Furthermore, while neural network models are capable of composing skills for solving new tasks, they fail to decompose a task into elementary skills that they use for solving new tasks.

The problem of integrating compositional computation is fundamental, it is not limited to task decomposition but it extends to learning generalized representations of concepts and organizing skills following a compositional structure. By analyzing aspects of human intelligence, compositionality can be regarded as a computational paradigm that emerges from the brain's structure and learning

experience. Following a research direction that focuses on incorporating various aspects of high-level brain function to reach human intelligence, I propose a framework for building brain-inspired neural networks. This framework describes principles of neural network architecture design and training. I use this framework to develop AbstractNet, a recurrent modular architecture with a controller that coordinates routing, module activation, and task progress. Preliminary experiments on this model show its capacity for multi-tasking, adapting computation time and routing strategies to task requirements. This work shows promising results for the development of deep learning models that emulate human intelligence. In this chapter, I will discuss these contributions and the general implications of this research for cognitive science and artificial intelligence.

5.1 Contributions

Compositionality has been a central concept in neuroscience and artificial intelligence research for decades. In the early decades of AI research, compositionality was an explicit feature of many systems since it was the basis for symbolic architecture. Today, more liberal uses of the term can be found in the field. Compositionality can be a feature of the data and representations, a property of the model [Chang et al., 2019, Ringstrom, 2022], or a nature of computation [Kurth-Nelson et al., 2023, Lake et al., 2015, Ellis et al., 2020]. Even though these notions eventually refer to the same idea of global compositionality, their applications remain more general. For example, the representations concern inputs of various types, such as images, 3D shapes, tasks, and programs, among others. My research is built on the hypothesis that frames compositionality as a computational basis for efficient learning and generalization. The contributions in this dissertation support these hypotheses by studying compositional learning and sample efficiency in deep learning models and humans. Furthermore, this dissertation proposes a general framework for developing brain-inspired architectures based on aspects of human intelligence.

To study compositional learning and sample efficiency, I took visual reasoning as a test bed, given the explicit compositional nature of this task. Taking inspiration from visual cognition theories [Ullman, 1987], I built a novel benchmark, named CVR, which addresses many limitations in existing challenges; it incorporates a variety of synthetic visual reasoning tasks that vary in the relationship between objects and the scene structures instantiated in each problem; the tasks

built in the benchmarks are built as compositions of other tasks, thus allowing for an evaluation and differentiation of model strengths at tasks and its capacity at composing skills; and it incorporates the sample efficiency as a measure of model performance.

Following extensive experiments, including single-task and multi-task training, curriculum, and reverse curriculum settings, on a host of baselines, including standard vision models and visual reasoning models, the analysis confirmed many hypotheses on compositional learning and sample efficiency. The findings revealed that even the most advanced pre-trained neural architectures necessitate significantly more training samples than humans to achieve comparable accuracy, aligning with prior research on sample efficiency [Lake et al., 2015]. Interestingly, our evaluation indicated that current neural architectures struggle to learn certain tasks, even with abundant samples and extensive prior visual experience. These results underscore the critical need for the development of more data-efficient and vision-oriented neural architectures to attain human-level artificial intelligence. Additionally, we investigated the models' generalization abilities across various rules, ranging from elementary rules to compositions and vice versa. Convolutional architectures displayed advantages in learning all visual reasoning tasks jointly and transferring acquired skills during training on elementary rules. However, they faced challenges in systematically generalizing from compositions to individual rules, suggesting that these architectures cannot decompose visual tasks into their fundamental components.

Incorporating aspects of human intelligence into neural networks has been a significant challenge in AI research. Examining the case of compositionality, there have been many attempts to implement compositional neural networks explicitly and implicitly, using special architectures, such as recursive computation or modularity, and training schemes. However, these attempts provide solutions to specific instantiations of compositionality that do not necessarily generalize to other instantiations. For example, a model can be specialized in learning the compositional structure of visual scenes and does not generalize to language phrases. Building a general model that leverages compositional computation is non-trivial. In this dissertation, I propose a research framework for pursuing this objective by taking inspiration from brain function. This framework defines human intelligence as the capacity to understand and perform tasks accurately and efficiently in terms of time, energy, computation, and experience. These factors of intelligence are attributed to innate capabilities such as the brain's structure and agency (the

ability to interact and control the environment) and emergent capabilities such as compositionality, environment simulation, and meta-learning. Compositionality emerges as a result of learning from interactions with the environment, which provides continuous, sequential, and multi-modal inputs with inherent compositionality and redundancy. Guided by this framework, the design of the neural architecture draws upon the brain's structural priors and its learning experience. The biological priors, including memory, attention, control, and predictive learning, have inspired AI researchers to develop general and specialized neural network architectures. However, relying solely on biological priors is insufficient for building an intelligent system; the learning experience plays a pivotal role in shaping the system's intelligence. Human experience exhibits compositionality and a curricular nature, which are taken into account in this framework. Uniting modularity, adaptive computation, recurrence, a variety of inductive biases, and curriculum learning, the proposed neural network architecture, AbstractNet, serves as an example and proof of concept for these ideas. Initial experiments demonstrate the model's proficiency in learning routing schemes, internal computation decisions, and distinct module functions in an end-to-end fashion. The model exhibits the ability to multitask in both homogeneous and heterogeneous tasks, adapt its computation time, and flexibly manipulate modules to perform various functions. AbstractNet is competitive with task-specific baselines and Universal Transformer, a self-attention-based monolithic architecture, in a restricted experimental setup. These results showcase the potential of the framework for constructing intelligent neural architectures capable of emulating aspects of human intelligence.

5.2 Limitations

While my work explores general aspects of compositionality and design principles for building brain-inspired models that emulate human intelligence, many questions remain unanswered about compositional learning in deep learning models. Furthermore, the model that I have proposed remains proof of the concept of the framework's usefulness and the feasibility of the approach. The model does not incorporate all aspects of the framework and has not been tested on compositional learning and generalization. In this section, I discuss a few limitations of the present investigation and ideas for future work.

Although CVR already contains a substantial array of visual relations, there is room for further improvement, particularly in the utilization of elementary vi-

sual relations. For instance, enhancing the generation of shapes parametrically based on specific geometric features could enhance the benchmark's effectiveness. Future work could expand CVR by incorporating additional relations borrowed from more specialized challenges, such as occlusion [Kim et al., 2019], line tracing [Linsley et al., 2018], and physics-based relations. Currently, the benchmark's rules are confined to 2 or 3 levels of abstraction, providing a systematic evaluation of the nine relations. Future work could explore hierarchical compositions of complex tasks to extend the benchmark. Although several vision models are included in the baseline, this work does not evaluate models that are built with a compositional inductive bias due to the necessity of adapting the benchmark's task framework, which could hinder the fairness of the evaluation. Another limitation of our work is the lack of a human baseline for compositional learning. However, recent behavioral work provides results that support our hypothesis on compositional learning in humans following the curriculum learning experimental protocol [Dekker et al., 2022]. To enhance the evaluation process, our methods for sample efficiency and compositionality could be refined and adapted to different scenarios. The sample efficiency score, for instance, is an empirical metric solely used for evaluating our benchmark, necessitating training all models on all data regimes for consistent scoring. While our work aligns with others in addressing sample efficiency, our primary objective is to encourage the development of more sample-efficient and general models in the field.

The proposed framework aims to construct brain-inspired models based on a single model, enriched with diverse inductive biases, to efficiently represent a comprehensive system. While our model shows promise in addressing compositionality and meta-learning, its inference efficiency may not be optimal. The current inference strategy involves controller intervention at each step, leading to slower execution—a departure from the brain's strategy of fast learning and slow inference, reminiscent of cortical learning in the CLS framework. Although AbstractNet can simulate such a learning strategy within individual module weights, its limited capacity to add connections between modules beyond routing poses challenges for efficient controller-free operations in complex tasks involving multiple modules.

While the framework assumes the feasibility of constructing curricula for tasks, this assumption may not hold for all tasks that the model is trained on, especially natural tasks. Decomposing tasks into constituent sub-tasks demands significant manual effort, leading to potential biases and limiting the exploration of alter-

native task decompositions. Benchmarking models trained on intricate curricula present challenges that have not been sufficiently addressed within the framework. This calls for careful consideration of both progress within the curriculum and the experience employed in the evaluation process.

The experiments presented provide a partial exploration of AbstractNet, focusing on modularity, module routing, and task interaction decisions, omitting module activation decisions, top-down modulation, and predictive modules. The diversity of modules remains limited, as we did not investigate hypernets, RNNs, or attention-based abstract modules. Additionally, crucial aspects such as meta-learning, compositionality, and out-of-distribution generalization were not included in the experimental evaluation. Furthermore, the experiments were confined to toy tasks and small-scale scenarios, revealing challenges in more intricate tasks like meta-reinforcement learning due to training instability and limited post-convergence exploration. We believe tailored curricula and regularization techniques could address these learning issues with future advancements. While AbstractNet displays a modest performance in this study's experimental setup, it does not outperform other baselines. Further investigations of its learning schemes and augmentations with design principles could potentially improve its performance, efficiency, and generalization.

This work serves as a pioneering conceptual exploration intended to ignite a new methodology in the development of brain-inspired neural network models with the ultimate goal of achieving human-level intelligence. The experiments offer a prototype and tangible proof of concept for the proposed model. Furthermore, our ongoing research delves into diverse avenues for further development within this framework, with the hope of inspiring fellow researchers to build upon and extend these innovative ideas.

5.3 Future Outlook

The AI field has tremendously evolved in the past few years, with a strong focus on larger scales and fine-tuning pre-trained models as keys to building more general models. The trends that we have witnessed thus far indicate that despite the impressive capabilities of these large models, they still lack the flexibility of compositional generalization of human intelligence. Given the fast development in the fields, we can only speculate on what will drive their development in the

future.

From the perspective of artificial intelligence, the architectural and training innovations proposed in this study may offer valuable insights even if other architectures demonstrate more flexible behavior in richer training regimes. These innovations have the potential to foster flexibility at more feasible data scales, although the data requirements for achieving human-like flexibility through brute force remain unclear and may be challenging to attain.

In conclusion, this work presents a computational perspective on how artificial intelligence could converge towards human-level intelligence. The future holds promise for unveiling new perspectives, and I am enthusiastic about the potential inspiration my research may provide for these upcoming endeavors.

CHAPTER 6

SUMMARY IN FRENCH

L'objectif ultime du domaine de l'IA est de construire des machines pouvant atteindre ou dépasser l'intelligence humaine. Cet objectif remonte à l'invention de l'ordinateur, où Alan Turing a déclaré, "Ce que nous voulons, c'est une machine capable d'apprendre par expérience" lors d'une conférence en 1947. Ce défi s'est avéré difficile étant donné l'infinité des possibilités et l'ampleur de l'objectif. En effet, le domaine a connu des décennies de tentatives innombrables et de cycles de changement d'intérêt pour les orientations de la recherche. Les chercheurs en sont venus à résoudre le problème en fixant des objectifs atteignables, tels que la reconnaissance de chiffres pour résoudre des "captchas" ou jouer aux échecs. La tendance à la simplification a entraîné la subdivision du domaine en plusieurs sous-domaines et le développement de systèmes spécialisés. Bien que cette tendance ait abouti à un biais au sein du domaine en faveur de la construction de systèmes spécialisés, elle constituait une étape inévitable vers la réalisation de l'objectif plus large de l'intelligence humaine. En conséquence, au cours des dernières décennies, le domaine de l'intelligence artificielle a fait d'énormes progrès sur plusieurs tâches, et les systèmes hautement spécialisés du passé sont progressivement remplacés par des systèmes de plus en plus généraux. Les grands modèles de langage [OpenAI, 2023], par exemple, ont réalisé des progrès substantiels dans le traitement du langage naturel, pouvant résoudre une variété de tâches avec une grande précision. De tels modèles à grande échelle [Bubeck et al., 2023, Ramesh et al., 2022] présentent un important potentiel pour atteindre le niveau d'intelligence humaine.

Bien que les réseaux neuronaux profonds (DNN) aient réalisé de nombreux développements vers l'intelligence humaine, ils restent moins flexibles et efficaces comparés au cerveau. La flexibilité et l'efficacité du cerveau se démontrent dans l'apprentissage de nombreuses compétences, comme la capacité de comprendre des tâches à partir de leurs descriptions, d'utiliser les connaissances acquises pour résoudre de nouvelles tâches, d'apprendre à partir de peu d'exemples et d'analyser son propre comportement. En revanche, les modèles d'apprentissage profond nécessitent de grandes quantités de données pour apprendre une tâche et généralisent mal aux nouvelles tâches ou aux changements de statistiques des données. Ces lacunes des DNN montrent qu'ils manquent de composants cruciaux pour atteindre le niveau d'intelligence humaine.

Naturellement, plusieurs chercheurs ont utilisé le cerveau comme source d'inspiration pour développer des systèmes plus intelligents à de nombreux niveaux de granularité, du niveau neuronal aux mécanismes de mémoire et d'attention, en passant par les fonctions cognitives telles que le contrôle et la simulation. Bien que le fonctionnement réel du cerveau reste énigmatique, cette approche repose sur des théories qui tentent d'expliquer sa fonction. Néanmoins, c'est une stratégie convaincante et sensée étant donné qu'elle a connu une grande popularité au cours des dernières décennies avec des degrés variables de succès et d'échec. De plus, les systèmes d'IA modernes dépassent les humains en terme de performance, mais échouent en termes de robustesse, d'efficacité d'apprentissage et de généralisation. Le chemin le plus prometteur pour améliorer les modèles d'IA sur ces aspects consiste à s'inspirer d'un système qui excelle en eux.

Étant donné la complexité du cerveau et notre compréhension limitée de sa fonction, il est important d'identifier d'abord les principes qui caractérisent son intelligence, de les relier aux mécanismes de sa fonction, puis de les traduire en principes de conception de DNN qui guident leur mise en œuvre. Dans cette thèse, je me concentre sur la compositionnalité, que je caractérise comme un aspect clé de l'intelligence humaine. Le principe de la compositionnalité a été utilisé pour caractériser le langage et la pensée [Frege, 1980, Fodor, 1975] en affirmant que le sens de l'ensemble est fonction de ses composants et de leur structure. Ce principe apparaît dans de nombreuses disciplines scientifiques, y compris l'apprentissage profond, où il est utilisé pour concevoir des architectures neuronales, évaluer des modèles et créer des tâches. Cependant, son application n'a pas réussi à combler avec succès l'écart entre les cerveaux et les machines.

Le premier chapitre situe mon travail au sein du domaine général en explorant

les caractéristiques de l'intelligence humaine, puis en discutant de la définition et de la manifestation de la compositionnalité dans la fonction du cerveau et les modèles d'apprentissage profond.

Legg and Hutter [2007b] propose une définition et une formalisation de l'intelligence artificielle : « L'intelligence mesure la capacité d'un agent à atteindre des objectifs dans de nombreux environnements. » Un aspect important de cette définition est la variété d'objectifs et d'environnements, ce qui souligne la robustesse de l'agent et sa capacité à transférer des connaissances entre les environnements. La définition plus récente de Chollet [2019] décrit l'intelligence comme « une mesure de l'efficacité de l'acquisition de compétences sur une gamme de tâches, par rapport aux connaissances antérieures, à l'expérience et à la difficulté de généralisation ». Cette dernière définition étend la précédente en y ajoutant l'efficacité de l'acquisition de compétences et contextualise sa mesure avec des connaissances antérieures sur l'agent et l'environnement.

Bien que la plupart des définitions offrent une représentation correcte de l'intelligence, elles décrivent l'intelligence par ses conséquences ; la définition est basée sur ce qu'un système intelligent est capable de faire, et non sur le processus qui produit le résultat. Prendre en compte ce dernier aspect comme base apporte une perspective différente de l'intelligence. Une conclusion générale de la plupart des définitions est que l'intelligence caractérise les processus mentaux derrière les actions, l'exécution de tâches et l'acquisition de compétences, pas seulement leur exécution. Les processus mentaux peuvent être considérés comme des opérations de traitement de l'information. De ce point de vue, l'intelligence peut être caractérisée par « l'organisation et la manipulation efficaces et productives de l'information ».

Lorsqu'un système interagit avec son environnement, il reçoit des informations sous des formes qu'il est prédisposé à traiter en suivant ses connaissances antérieures, et il choisit des actions. Selon la structure de l'environnement, l'intelligence du système peut être mesurée par son succès à accomplir des tâches depuis sa création et tout au long de sa durée de vie au sein de l'environnement. Un système parfait résout toutes les tâches tout au long de son existence dans l'environnement et n'a pas besoin d'apprentissage. Dans un scénario pratique, un système ne peut pas résoudre les tâches dès son initialisation. Sa capacité à résoudre les tâches dépend de sa compréhension des tâches et de l'acquisition de compétences à partir de l'expérience, ce qui dépend fortement de sa compétence en traitement de l'information. Étant donné les ressources de calcul limitées du système, son effi-

cacité à manipuler l'information détermine son niveau d'intelligence. Selon cette définition, l'apprentissage peut être considéré comme une forme d'efficacité, car il réduit le temps et les ressources utilisés pour résoudre une tâche. La créativité, qui consiste en la combinaison significative de connaissances antérieures pour générer de l'information, est également un aspect important de l'intelligence, car elle peut être un outil pour trouver des solutions efficaces à des problèmes nouveaux. L'abstraction peut être considérée comme un exemple de la créativité du système pour la généralisation. Dans l'ensemble, l'intelligence d'un système n'est pas mesurée uniquement par sa performance finale sur une tâche, elle dépend également de :

- L'efficacité d'apprentissage : la quantité d'expérience nécessaire au système pour atteindre sa performance maximale sur une tâche.
- L'efficacité temporelle : le temps moyen passé à résoudre une instance de tâche.
- L'efficacité des ressources : les ressources de calcul utilisées pour résoudre la tâche.
- L'efficacité énergétique : la quantité d'énergie consommée lors de la résolution de la tâche.

En analysant ces aspects dans le contexte d'un environnement et d'un système donnés, on peut déduire les caractéristiques d'un comportement intelligent.

De nombreux aspects de notre environnement déterminent les caractéristiques de l'intelligence. Les humains vivent dans un environnement hautement complexe, dynamique et ouvert, avec un accès partiel à ses informations. Un aspect important de notre environnement est sa nature compositionnelle. Les éléments de l'environnement, c'est-à-dire les objets avec lesquels nous interagissons, sont construits de manière hiérarchique en tant que compositions de leurs composants. Dans cet environnement, les humains ont également des limites et des contraintes en termes d'énergie, de temps et de ressources de calcul.

L'intelligence humaine dans ce contexte dépend des prédispositions physiques pour l'analyse des informations ; les sens et les connaissances antérieures intégrées à la structure du cerveau au cours de l'évolution, ainsi que la capacité à apprendre de l'expérience. Confrontée à des informations de bas niveau très

complexes avec des tâches définies à des niveaux élevés d'abstraction, la première caractéristique de l'intelligence humaine est l'*abstraction* : la capacité à organiser l'information en filtrant les variables non pertinentes pour la tâche et en reconnaissant les schémas extraits des expériences passées. L'accès partiel aux informations force l'émergence de nombreuses stratégies pour déduire des informations pertinentes pour la tâche; explorer l'environnement, utiliser davantage d'expérience pour l'apprentissage ou déduire des inconnues en fonction des informations disponibles et des expériences passées. Ces stratégies sont sélectionnées selon leur efficacité. Les humains déduisent les informations inconnues en construisant un modèle spécifique à la tâche de l'environnement. C'est une caractéristique importante de l'intelligence humaine, car *la construction de modèles* est utilisée à de nombreuses fins ; planification des actions, utilisation de la simulation pour construire des hypothèses et génération d'expériences pour l'apprentissage. Compte tenu de la grande complexité de l'environnement, l'inférence sur certaines tâches peut solliciter les ressources limitées du système. Les humains compensent la précision sur ces tâches par l'efficacité dans l'utilisation de leurs ressources de calcul en développant des solutions approximatives aux problèmes d'inférence. *L'inférence approximative rapide* est également une caractéristique importante de l'intelligence dans un système aux ressources limitées.

Malgré les progrès considérables dans ces divers sous-domaines, les réseaux neuronaux restent limités. Ils sont considérés au mieux comme de bons modèles pour les capacités d'inférence rapide chez les humains. Par exemple, les architectures de convolutives [Krizhevsky et al., 2017, He et al., 2015] modélisent la reconnaissance des objets dans le cortex visuel, un processus qui est supposé impliquer principalement la propagation des informations visuelles pour extraire les catégories d'objets [Eberhardt et al., 2016, Yamins et al., 2014, Rajalingham et al., 2015]. Les réseaux neuronaux souffrent d'un entraînement lent et inefficace en termes de données, d'un manque de robustesse dans des contextes hors distribution [Geirhos et al., 2020a], de biais envers les tendances statistiques dans les données, d'oubli catastrophique et de manque de généralisation compositionnelle. Ces limitations pourraient être causées par de nombreux facteurs ; le manque de flexibilité dans les calculs des réseaux neuronaux, la disparité dans les stratégies d'apprentissage et l'expérience entre les cerveaux et les réseaux neuronaux. Les modèles a grande échelle tels que les grands modèles de langage (LLM) [Brown et al., 2020, Touvron et al., 2023a,b] et les grandes architectures multimodales [OpenAI, 2023, Driess et al., 2023] abordent un bon nombre de ces problèmes grâce aux grandes quantités de paramètres et de données

d'apprentissage, mais à des coûts de calcul et d'énergie exorbitants. Cependant, leurs capacités restent limitées par rapport aux humains [Kaddour et al., 2023] car ils montrent des performances inférieures à celles des humains sur de nombreuses tâches de raisonnement logique et produisent des résultats peu fiables, car leurs résultats varient fortement en fonction de leurs entrées.

Une hypothèse populaire dans le domaine attribue le manque de fiabilité, de généralisation et de flexibilité à leur incapacité à implémenter le calcul compositionnel. En raison de la nature compositionnelle de notre environnement, on pense que les humains utilisent la compositionnalité comme base pour la représentation et le calcul. Ces idées sont partagées par d'autres travaux (tels que Lake et al. [2016]) qui notent la compositionnalité comme une caractéristique importante de l'intelligence humaine. Smolensky et al. [2022] attribue l'intelligence humaine à la continuité et à la compositionnalité dans le calcul neuronal, tandis que l'absence de continuité explique l'échec des premiers systèmes d'IA symbolique, et l'absence ou le manque de compositionnalité explique l'échec des systèmes modernes basés sur les réseaux neuronaux.

La compositionnalité a été un concept central dans la recherche en IA depuis des décennies. Dans les débats sur le connexionnisme et la structure symbolique de l'architecture cognitive, les réseaux neuronaux, en tant que systèmes connexionnistes, ont été critiqués pour leur absence de manipulation symbolique compositionnelle [Fodor and Pylyshyn, 1988, Lake et al., 2016, Lake and Baroni, 2018, Marcus, 2018]. De nombreuses études ont testé la capacité des réseaux neuronaux à résoudre des tâches nécessitant une généralisation compositionnelle, avec des résultats mitigés [Christiansen and Chater, 1994, Marcus, 1998, Botvinick and Plaut, 2006, Bowers et al., 2009, Botvinick and Plaut, 2009, Frank et al., 2009, Bowman et al., 2015, Frank, 2014]. Des tentatives ont également été faites pour développer un schéma de représentation des structures compositionnelles à l'aide de vecteurs [Smolensky, 1990]. Les idées sur la compositionnalité ont été adoptées pour expliquer des modalités au-delà du langage et de la pensée. Par exemple, Hoffman and Richards [1984], Biederman [1985] ont théorisé que le système visuel décompose les objets en leurs parties. Ces dernières années, le domaine s'est largement développé dans de nombreuses directions en explorant ; le développement de formalisations pour évaluer la compositionnalité dans les réseaux neuronaux, l'analyse des modèles pour leur structure compositionnelle et l'amélioration de la généralisation compositionnelle grâce à de nouvelles architectures ou à des schémas d'entraînement spéciaux.

La première idée à clarifier dans cette question de recherche est la nature de la compositionnalité dans les réseaux neuronaux. Les premières recherches sur les tâches visuelles, notamment la classification visuelle, ont mis en avant les réseaux convolutifs profonds (CNN). Les CNN sont depuis devenus les modèles standard en vision et leur succès a été en partie attribué à leur capacité à extraire des motifs hiérarchiquement à partir d'images. Cette hiérarchie de motifs a été considérée comme une caractéristique de la compositionnalité [Zeiler and Fergus, 2013]. Bien que les CNN aient une structure qui représente des concepts à différents niveaux d'abstraction, cette structure reste limitée à la représentation des motifs de l'image. Par exemple, les CNN standard ne peuvent pas décomposer une scène en objets et en composants. Ainsi, les CNN peuvent être considérés comme possédant une structure compositionnelle, mais une structure restreinte dans sa capacité de représentation et d'utilité pour d'autres tâches. Une structure compositionnelle peut également être intégrée aux données et au processus, comme dans les réseaux neuronaux récursifs [Socher et al., 2013]. Ces exemples soulèvent une question importante : qu'est-ce que la compositionnalité caractérise dans les réseaux neuronaux, les représentations ou la structure du modèle ? Est-ce implicite ou explicite ? Et dans quelle mesure est-elle généralisable ?

Étant donné la difficulté d'organiser des données diverses dans un format compositionnel, le domaine s'est concentré sur l'étude de la compositionnalité implicite dans les représentations et les paramètres des architectures neuronales, et sur l'intégration de la compositionnalité dans la structure interne du modèle. La généralisation compositionnelle a été étudiée dans divers contextes : l'apprentissage sans données (zero-shot leanring) dans la vision [Yang et al., 2020, Mancini et al., 2021, Misra et al., 2017, Naeem et al., 2021, Purushwalkam et al., 2019, Atzmon et al., 2020, Wang et al., 2020], les représentations 3-dimensionnelles [Tulsiani et al., 2018], le raisonnement visuel [Johnson et al., 2017a], l'apprentissage par renforcement [Gur et al., 2022], le langage [Lake and Baroni, 2018, Keysers et al., 2020] et des tâches abstraites telles que les mathématiques [Saxton et al., 2019]. Dans la plupart des contextes, les modèles sont évalués en termes de systématité, où de nouvelles combinaisons des concepts déjà rencontrés pendant l'entraînement sont introduites lors des tests. Les résultats de ces études varient, avec une tendance à l'échec des modèles standard en termes de généralisation compositionnelle.

Prenons l'exemple de la navigation basée sur le langage ; les résultats montrent que les modèles récurrents standards n'apprennent pas de manière compo-

sitionnelle [Loula et al., 2018, Lake and Baroni, 2018] et bien que les modèles de langage pré-entraînés masqués aient de meilleures performances [Furrer et al., 2021], ils n'apprennent toujours pas de manière compositionnelle. Cependant, Lake and Piantadosi [2019] montre que l'ajout d'une mémoire à une architecture seq2seq lui permet de résoudre de nombreux tests SCAN.

Étant donné que la plupart des tests de généralisation compositionnelle se limitent à un aspect de la compositionnalité, qui est la systématicité, Hupkes et al. [2020] propose PCFG SET, un ensemble de tests pour 5 aspects de la compositionnalité : systématicité, productivité, localité vs globalité, substitutivité et surgénéralisation. Leur analyse sur les architectures standard montre qu'elles échouent à la plupart des tests.

Les améliorations de la compositionnalité des modèles de réseaux neuronaux varient en fonction des méthodes d'entraînement et de l'utilisation de différents biais inductifs. Baan et al. [2019], Hupkes et al. [2019] montrent que l'entraînement de modèles avec des biais basés sur l'attention pousse les modèles à mettre en œuvre des solutions plus compositionnelles et améliore la généralisation compositionnelle. Dans une tâche d'apprentissage par renforcement, Hill et al. [2020] montrent qu'augmenter la variété perceptive et le réalisme de l'environnement améliore la généralisation compositionnelle dans le langage. Ces exemples montrent que l'expérience d'entraînement influence le comportement compositionnel des réseaux neuronaux et peut même orienter les modèles vers l'apprentissage de stratégies de calcul compositionnel.

Les architectures modulaires ont été utilisées pour implémenter explicitement des calculs compositionnels dans de nombreux scénarios [Andreas et al., 2016a, Hu et al., 2017]. Certaines approches utilisent l'induction de programmes avec des primitives de programme [Johnson et al., 2017a]. Ces approches nécessitent la mise en œuvre d'un ensemble diversifié de primitives de programme et sont limitées dans leur capacité à apprendre de nouveaux programmes. Des inspirations tirées du fonctionnement du cerveau ont été utilisées pour implémenter d'autres approches [Russin et al., 2019], mais elles présentent des améliorations limitées par rapport aux architectures standard.

Les architectures modernes d'apprentissage en profondeur à grande échelle, telles que les modèles génératifs [Ramesh et al., 2022, Rombach et al., 2022], les grands modèles de langage [Brown et al., 2020, Touvron et al., 2023a,b] et les modèles de fondation multimodaux [OpenAI, 2023, Driess et al., 2023, Yu et al.,

2022], présentent des capacités impressionnantes dans diverses tâches en contexte de zéro-shot. Ils semblent présenter l'échelle comme une solution appropriée pour la généralisation. Cependant, au-delà des coûts exorbitants d'entraînement et d'inférence, leurs échecs démontrent la fragilité et les performances sous-humaines sur de nombreuses tâches de raisonnement. Leurs limitations en matière de compositionnalité ont été démontrées dans divers scénarios [Dziri et al., 2023].

Cet aperçu succinct de la recherche sur la compositionnalité met en évidence qu'à ce jour, il n'existe pas de modèles d'apprentissage en profondeur qui se comportent de manière fiable dans les tests de généralisation compositionnelle. Les architectures actuelles reposent sur des biais inductifs uniques qui limitent leur expressivité et leur capacité à représenter des structures compositionnelles diverses. De plus, les tests de compositionnalité évaluent principalement la compositionnalité du système à l'inférence. À notre connaissance, les modèles n'ont pas été évalués en termes de leur capacité à décomposer les tâches lors de l'apprentissage. L'un des objectifs de cette thèse est d'étudier cet aspect de l'apprentissage dans les réseaux neuronaux. Le chapitre 2 détaille le développement de CVR, un test de raisonnement visuel compositionnel, qui comprend 103 tâches construites en composant de 9 relations visuelles élémentaires. Ce test est développé pour évaluer l'apprentissage compositionnel dans les réseaux neuronaux. Pour le construire, je propose une nouvelle méthode pour créer des problèmes de raisonnement visuel en mettant en avant la compositionnalité.

Le raisonnement visuel est une capacité complexe nécessitant un haut niveau d'abstraction sur une entrée sensorielle de haute dimension. Il met en évidence la capacité des humains à manipuler des concepts et des relations en tant que symboles extraits de l'entrée visuelle. L'efficacité avec laquelle les humains apprennent de nouveaux concepts et relations visuelles, comme l'illustrent l'intelligence fluide et les tests de raisonnement non verbal, est tout aussi fascinante. Pour ces raisons, j'ai choisi le raisonnement visuel comme un test pour évaluer l'apprentissage compositionnel et l'efficacité en terme de données d'apprentissage.

Seuls quelques référentiels abordent ces aspects de l'intelligence humaine dans le raisonnement visuel. L'un de ces référentiels, ARC [Chollet, 2019], propose des problèmes variés de raisonnement visuel. Cependant, le peu d'exemples d'entraînement, seulement 3 par tâche, rend le test difficile pour toutes les méthodes, en particulier les réseaux neuronaux. D'autres référentiels ont conduit au développement de nouveaux modèles basés sur des réseaux neuronaux qui comblent des lacunes spécifiques entre l'intelligence humaine et artificielle [Barrett et al.,

2018, Zhang et al., 2019, Fleuret et al., 2011]. Certains se concentrent sur l'évaluation des exigences perceptuelles de la tâche [Fleuret et al., 2011], qui incluent la détection de caractéristiques, la reconnaissance d'objets, le regroupement perceptuel et le raisonnement spatial. D'autres évaluent les exigences de raisonnement logique [Barrett et al., 2018, Zhang et al., 2019], telles que le raisonnement symbolique, les analogies et le raisonnement causal. Cependant, ils manquent de la variété des relations abstraites présentes dans la scène, ou de la variété sémantique et structurelle des scènes sur lesquelles ils instancient ces relations abstraites.

Créer de nouvelles tâches de raisonnement visuel peut être difficile. Dans ce référentiel, nous standardisons un processus de création de tâches de manière compositionnelle basé sur un ensemble élémentaire de relations et d'abstractions. Ce processus nous permet d'exploiter un large ensemble de relations visuelles ainsi que de règles abstraites, rendant ainsi possible l'évaluation à la fois des exigences perceptuelles et logiques du raisonnement visuel. La nature compositionnelle des tâches offre une opportunité d'étudier les stratégies d'apprentissage utilisées par les méthodes existantes. CVR s'appuie sur des référentiels d'IA antérieurs [Fleuret et al., 2011, Chollet, 2019] et s'inspire d'une littérature en sciences cognitives [Ullman, 1987] sur le raisonnement visuel. Dans la suite, nous décrirons le processus de génération des exemples du test.

Chaque exemple de test est constitué de quatre images, dont une est l'intrus. L'intrus est choisi selon une règle spécifique. Chaque image contient une scène composée de plusieurs objets. Un objet est défini comme un contour fermé avec un ensemble d'attributs : la forme, la position, la taille, la couleur, la rotation et la direction. D'autres attributs décrivent la scène ou les relations de bas niveau entre les objets. Le dénombrement correspond au nombre d'objets, de groupes d'objets ou de relations. L'intériorité indique qu'un objet contient un autre objet à l'intérieur de son contour. Le contact indique que deux contours d'objet se touchent. Ces 9 attributs constituent la base des neuf relations élémentaires.

CVR intègre 103 règles de référence uniques, comprenant 9 règles instanciant les neuf relations visuelles élémentaires et 94 règles supplémentaires construites sur des compositions des relations. Ces compositions couvrent toutes les paires de règles élémentaires et incluent jusqu'à 4 relations. Bien que certaines règles soient composées des mêmes relations élémentaires, elles restent uniques dans leur structure de scène ou leurs associations avec d'autres relations. 20 relations sont des compositions de relations élémentaires uniques, 65 sont des compositions d'une paire de relations et 9 sont des compositions de plus de 2 relations

élémentaires. La génération procédurale d'échantillons de problèmes nous permet de créer un nombre arbitraire d'échantillons. Ce jeu de données comprend 10 000 échantillons de problèmes d'entraînement, 500 échantillons de validation et 1 000 échantillons de test pour chaque tâche. De plus, un ensemble de tests de 1000 échantillons est fourni pour évaluer la généralisation hors distribution.

Le chapitre 3 se concentre sur l'évaluation des modèles d'apprentissage en profondeur sur le dataset de CVR. Cette évaluation se concentre sur les modèles de raisonnement visuel abstrait de pointe et sur les modèles de vision standard. Ces modèles ont atteint des performances élevées sur plusieurs tâches de raisonnement visuel dans des travaux précédents [Wu et al., 2020, Vaishnav et al., 2022], mais ils nécessitent toujours de grandes quantités de données.

L'évaluation comprend des expériences à grande échelle qui couvrent une multitude de configurations, notamment l'entraînement multitâche et individuel, le pré-entraînement avec auto-supervision sur des images du jeu de données pour contraster l'apprentissage des représentations visuelles par rapport aux règles de raisonnement visuel abstrait, l'entraînement sur plusieurs quantités différentes de données, les tests de transfert d'apprentissage entre les tâches du jeu de données, et l'évaluation de la généralisation hors distribution. Je présente une analyse de la difficulté des tâches, qui fournit des informations sur les forces et les faiblesses des modèles actuels.

Nos résultats suggèrent que même les meilleures architectures neuronales pré-entraînées nécessitent plus d'échantillons d'entraînement que les humains pour atteindre le même niveau de performance, ce qui est cohérent avec des travaux antérieurs sur l'efficacité des échantillons [Lake et al., 2015]. Notre évaluation a également révélé que les architectures neuronales actuelles n'apprennent pas plusieurs tâches, même lorsqu'elles disposent d'une abondance d'échantillons et d'une vaste expérience visuelle antérieure. Ces résultats soulignent l'importance de développer des architectures neuronales plus efficaces en termes de données et orientées vers la vision. De plus, la capacité de généralisation des modèles est évaluée sur différentes règles, des règles élémentaires aux compositions et vice versa. Les résultats montrent que les architectures de convolution bénéficient de l'apprentissage conjoint de toutes les tâches de raisonnement visuel et du transfert des compétences acquises lors de l'entraînement sur les règles élémentaires. Cependant, elles ont également échoué à généraliser de manière systématique des compositions à leurs règles individuelles. Ces résultats indiquent que les architectures de convolution sont capables de transférer des compétences

entre les tâches, mais n'apprennent pas en décomposant une tâche visuelle en ses composants élémentaires. Les fortes demandes en quantités de données et la faible généralisation des réseaux neuronaux par rapport aux humains pourraient être dues à leur stratégie d'apprentissage non compositionnelle et au manque de curriculum dans leur entraînement. Cette idée est étayée par des preuves comportementales et computationnelles [Dekker et al., 2022] où il est démontré que les humains généralisent de manière compositionnelle au-delà des capacités des réseaux neuronaux. De plus, l'entraînement curriculaire améliore la généralisation, ce qui souligne l'importance d'introduire une organisation dans la difficulté des tâches d'entraînement.

Bien que notre travail aborde des questions importantes sur l'efficacité et la compositionnalité, nos méthodes d'évaluation pourraient être encore améliorées et adaptées à différentes configurations. Par exemple, le score qui quantifie l'efficacité en quantité de données est une mesure empirique utilisée uniquement pour évaluer notre test. Il faut entraîner tous les modèles sur tous les régimes de données pour que le score soit cohérent. Bien que notre travail ne soit pas unique en abordant l'efficacité des échantillons, son objectif est de promouvoir des modèles plus efficaces en termes d'échantillons et plus généraux.

Dans la littérature sur le raisonnement visuel, les modèles polyvalents tels que les ViTs et les CNN sont fournis en tant que références, avec des approches plus complexes reposant sur des biais inductifs supplémentaires pour le raisonnement tels que les RNN, les GNN et les « relation networks » [Johnson et al., 2017a, Santoro et al., 2017, Chen et al., 2021b]. Ces architectures atteignent des performances décentes, mais ont une mauvaise généralisation et une faible efficacité en quantité de données d'entraînement. Les solutions plus prometteuses pour le raisonnement visuel utilisent l'idée de modularité [Andreas et al., 2016b, Chen et al., 2021c, Hudson and Manning, 2018, 2019, Mittal et al., 2021, Rahaman et al., 2021, Goyal et al., 2019]. Les réseaux neuronaux modulaires sont composés d'un ensemble de modules qui effectuent différentes opérations. Ces modèles sont généralement orchestrés par un module de contrôleur qui exécute des instructions basées sur le langage. Je suppose que la modularité pourrait être un biais inductif fondamental pour la compositionnalité. Équipé d'un module de contrôleur approprié et de mécanismes de routage de l'information, un réseau modulaire pourrait manipuler de manière flexible de nouveaux concepts et construire des représentations contextuelles. Bien que ces modèles offrent l'avantage de l'interprétabilité et d'une meilleure généralisation OOD, ils sont notoirement difficiles à entraîner.

D'autres méthodes se concentrent sur la décomposition de scènes [Burgess et al., 2019, Engelcke et al., 2019, Li et al., 2020], ces modèles reposent sur l'attention et des représentations centrées sur les objets en tant que biais inductifs pour construire des représentations de scènes utiles pour le raisonnement visuel [Ding et al., 2021]. Dans une autre approche, certaines solutions font évoluer des architectures simples, basées sur des transformateurs et des convolutions, et s'appuient sur le pré-entraînement auto-supervisé pour obtenir des performances impressionnantes sur plusieurs tâches de vision par ordinateur multimodales [Ramesh et al., 2022, Yu et al., 2022]. Cependant, la capacité de ces modèles à exploiter la compositionnalité est limitée par leurs composants architecturaux ; les transformateurs et les ResNets. Je pense que la modularité, l'attention et la factorisation en objets sont des biais inductifs essentiels pour atteindre l'efficacité en quantité de données et la compositionnalité dans CVR. L'attention est utilisée pour extraire le graphe de scène de l'image, tandis que les modules mettent en œuvre différentes stratégies pour résoudre différentes tâches de raisonnement visuel. Je crois que les modèles futurs de raisonnement visuel devraient implémenter ces biais inductifs tout en s'inspirant de la cognition humaine pour orchestrer le raisonnement visuel comme une exécution de programme.

Les trois premiers chapitres se sont principalement concentrés sur la caractérisation de différentes facettes de l'intelligence, mettant en évidence l'importance de la compositionnalité en tant que paradigme computationnel pour l'apprentissage et la généralisation efficaces, et enquêtant sur les disparités entre l'intelligence des machines et celle des humains dans le raisonnement visuel. Pour réduire l'écart entre les humains et les machines en termes d'intelligence, une approche prometteuse consiste à s'inspirer du cerveau pour construire des systèmes d'IA. Conformément à cette vision, l'objectif de ce chapitre est d'introduire des principes de conception architecturale et des stratégies de formation qui s'inspirent du fonctionnement du cerveau.

Bien qu'il soit important de s'inspirer du cerveau, il est essentiel de trouver un équilibre entre la reproduction du cerveau et la simple mise en œuvre de ses fonctions de haut niveau. Je crois que reproduire chaque détail complexe du cerveau, de la dynamique des décharges neuronales à la structure anatomique, dans des modèles de réseaux neuronaux ne sera potentiellement pas nécessaire pour atteindre l'intelligence humaine. En réalité, essayer de créer un modèle précis du cerveau peut souvent conduire à une complexité inutile et à des inefficacités computationnelles. Mon objectif est plutôt de comprendre les principes et les mécan-

ismes fondamentaux qui contribuent à l'intelligence. En distillant ces principes dans les systèmes d'IA, même s'ils ne reflètent pas exactement la fonction du cerveau, ils pourraient afficher un comportement intelligent sans être alourdis par les complexités biologiques. Identifier les facteurs cruciaux de l'intelligence est une tâche complexe. Par exemple, la question de savoir si les neurones et leur dynamique doivent être implémentés avec précision ou si différentes unités de calcul peuvent capturer leur expressivité reste ouverte. De plus, discerner les propriétés qui émergent du système de celles qui sont innées présente d'autres défis. Par exemple, dans le premier chapitre, la compositionnalité est supposée être une propriété émergente du cerveau. Ainsi, je crois que pour qu'un système basé sur un réseau neuronal puisse efficacement implémenter et exploiter la compositionnalité, à l'instar des humains, il ne peut pas se reposer sur un seul biais inductif. Au lieu de cela, le système devrait apprendre à utiliser la compositionnalité comme un paradigme computationnel à travers l'expérience et l'apprentissage.

Isoler les caractéristiques de la fonction du cerveau qui contribuent à son intelligence reste un défi important. Cependant, nous pouvons identifier des principes clés de sa construction qui contribuent à des fonctions de haut niveau importantes ; la distinction des composants de traitement d'information, la variété de l'architecture et la spécialisation de chaque composant, la coordination des fonctions par un système exécutif au sein et entre les composants, l'adaptabilité et le contrôle de l'apprentissage, l'agence et l'expérience d'apprentissage structurée et variée. En utilisant ces principes avec les propriétés émergentes de l'intelligence humaine, je propose un ensemble de méthodes pour la conception et l'entraînement d'architectures de réseaux neuronaux.

Pour fournir une preuve de concept, je développe un modèle de réseau neuronal suivant ces méthodes ; AbstractNet, un réseau neuronal modulaire capable de contrôler ses calculs et de les adapter aux exigences des tâches. Les expériences préliminaires avec AbstractNet montrent sa capacité à résoudre de nombreuses tâches impliquant diverses compétences en apprenant à manipuler de nombreux modules de manière intégrée. Bien qu'AbstractNet montre des résultats prometteurs, il pourrait être encore amélioré en suivant davantage de principes de conception, tels que le contrôle à travers des connexions modulatrices, les modules prédictifs et le contrôle des signaux d'apprentissage.

Au sein du vaste domaine de l'IA, cette thèse aborde une caractéristique importante de la fonction cognitive que les modèles d'apprentissage profond ont du mal à incorporer dans la mesure atteinte par les cerveaux humains : le principe

de la compositionnalité. Plusieurs hypothèses postulent que la compositionnalité est un outil pour organiser la pensée, ce qui est démontré dans plusieurs aspects de la fonction cognitive tels que la représentation des connaissances et l'apprentissage des tâches. Plusieurs études examinent la compositionnalité dans les modèles d'apprentissage profond et tentent d'incorporer le calcul compositionnel en eux. Néanmoins, certains aspects de la compositionnalité, tels que la composition et la décomposition des compétences lors de l'apprentissage, doivent encore être explorés dans l'apprentissage profond. Je me concentre sur ces aspects de l'apprentissage et les relie à l'efficacité en quantité de données d'entraînement. J'ai développé un référentiel pour le raisonnement visuel compositionnel (CVR) qui évalue l'apprentissage compositionnel et l'efficacité de l'échantillonnage chez les humains et les réseaux neuronaux. Les expériences ont montré un écart important entre l'efficacité des humains et des modèles d'apprentissage profond préentraînés sur une variété de tâches visuelles. De plus, tandis que les modèles de réseaux neuronaux étaient capables de composer des compétences pour résoudre de nouvelles tâches, ils échouaient à décomposer une tâche en compétences élémentaires qu'ils utilisent pour résoudre de nouvelles tâches.

Le problème d'intégration du calcul compositionnel est fondamental, il ne se limite pas à la décomposition des tâches, mais s'étend à l'apprentissage de représentations généralisées de concepts et à l'organisation de compétences selon une structure compositionnelle. En analysant les aspects de l'intelligence humaine, la compositionnalité peut être considérée comme un paradigme computationnel émergent de la structure du cerveau et de l'expérience d'apprentissage. Suivant une orientation de recherche qui vise à incorporer divers aspects de la fonction du cerveau pour atteindre l'intelligence humaine, je propose un cadre pour construire des réseaux neuronaux inspirés du cerveau. Ce cadre décrit les principes de conception et de formation de l'architecture des réseaux neuronaux. J'utilise ce cadre pour développer AbstractNet, une architecture modulaire récurrente avec un contrôleur qui coordonne le routage, l'activation des modules et la progression des tâches. Les expériences préliminaires sur ce modèle montrent sa capacité à effectuer plusieurs tâches, à adapter le temps de calcul et les stratégies de routage aux exigences de la tâche. Ce travail présente des résultats prometteurs pour le développement de modèles d'apprentissage profond qui émulent l'intelligence humaine et j'espère qu'il inspirera d'autres recherches sur les architectures inspirées du cerveau.

Dans l'ensemble, le domaine de l'IA a grandement évolué au cours des dernières

années, en mettant un accent fort sur la mise à grande échelle des architectures simples et l'adaptation des modèles pré-entraînés comme clés pour la construction de modèles plus généraux. Les tendances que nous avons observées jusqu'à présent indiquent que malgré les capacités impressionnantes de ces grands modèles, ils manquent encore de la flexibilité de la généralisation compositionnelle de l'intelligence humaine. Étant donné le développement rapide dans les domaines, nous ne pouvons que spéculer sur ce qui stimulera son développement à l'avenir. Du point de vue de l'intelligence artificielle, les innovations architecturales et d'entraînement proposées dans cette étude peuvent offrir des aperçus précieux, même si d'autres architectures montrent un comportement plus flexible dans des régimes d'entraînement plus riches. Ces innovations ont le potentiel de favoriser la flexibilité à des échelles de données plus réalisables, bien que les exigences en matière de données pour atteindre une flexibilité semblable à celle de l'humain par la force brute restent floues et puissent être difficiles à atteindre.

En conclusion, ce travail présente une perspective computationnelle sur la façon dont l'intelligence naturelle et artificielle pourrait converger vers une intelligence de niveau humain. L'avenir promet de dévoiler de nouvelles perspectives, et je suis enthousiaste à l'idée de l'inspiration potentielle que mes recherches pourraient fournir pour ces entreprises à venir.

Appendix

CHAPTER 7

VISUAL REASONING EXPERIMENTS

A1 Experiment Details

RPM Baselines In order to provide a fair comparison for models designed for solving RPMs, we adapt the odd-one-out task to the matrix and choice selection task setup. In the RPM setting, models are fed the nine panels with tags that indicate the position on the matrix. Each of the eight choice panels is concatenated individually with the eight context panels. The model outputs a logit for each of the eight matrices used to compute the cross-entropy loss. The training process is explained in detail here Barrett et al. [2018]. We discard the position tags in our setting since the four images have no sense of progression. We replace context panels with the four problem images and use the same four images as choice panels, with the correct choice being the outlier.

Self-Supevised Pretraining The SSL pretraining objective function maximizes the similarity between transformations of the same image. When SSL is performed on datasets such as ImageNet Deng et al. [2009], where the downstream task is classification, the images are transformed in ways that maintain the class information. These transformations are augmentations, including spatial transformations such as random cropping and flipping, noise (e.g., additive Gaussian noise or blurring), and color transformations such as grayscale and color jittering. In our setup, we use the following augmentations: random resize, random Gaussian blur-

ring, random horizontal flips, and random rotations in multiples of 90 degrees. To ensure the variety of images covers all structures used in the dataset, we select one image from each problem in the dataset, for a total of 1 million images. This number of images is equivalent to the number of images in the ImageNet dataset, which is customarily used in SSL pretraining.

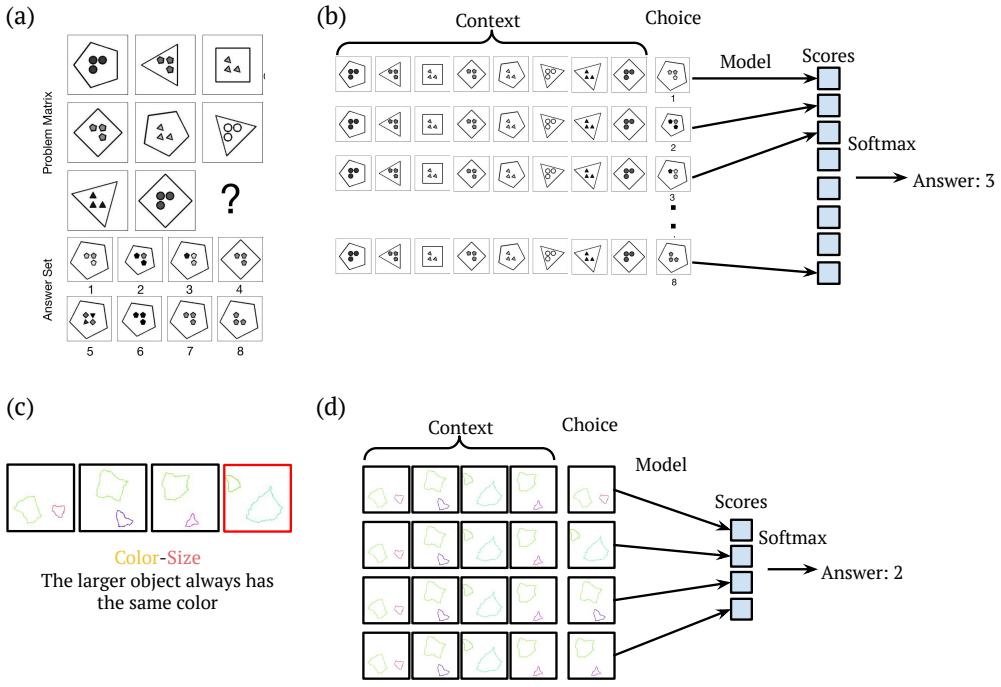


Figure A1: **RPM training setup**: (a) A sample RPM problem adapted from Zhang et al. [2019], the matrix contains context panels, and a choice is taken from the answer set. (b) Inference in RPM models: the model takes all context panels with one of the choices and outputs a score. These scores from 8 choices are used for computing the cross-entropy loss. (c) An odd-one-out problem based on size and color. (d) The problem is adapted to RPM by placing all images in context and choice. The odd-one-out has the highest score among the choices.

A2 Architectures and Hyperparameters

We adapt model architectures from reference implementations: ResNet-50[1], ViT [Chen et al., 2021a][2], SCL [3] and WReN [4]. We also endow SCL with ResNet18, a strong vision backbone. We name this architecture SCL-ResNet-18. All model architectures All models are trained using images with a size of 128×128 pixels. Figure A2 illustrates the architectures of all the baselines.

In preliminary experiments, we hand-tuned the learning rate and weight decay for all models and selected the hyperparameters that achieved the highest performance in the joint training setting for each model. We also analyzed the random seed effect and observed that training results are robust with respect to the seed.

We equip standard vision models, ViT-small and Resnet 50, with an MLP that extracts task-specific information. It takes as input image features and the task embedding and outputs a lower-dimensional vector used for computing the pairwise distances and the loss. The MLP contains 2 layers; the hidden layer's size is 2048, and the output size is 128. The task embedding space has 64 dimensions. All models were trained using Adam optimizer [Kingma and Ba, 2014] for 100 epochs, with early stopping after 30 epochs. The mini-batch size used for training all models is 64, except when the training set is smaller. The learning rates are scaled linearly with the batch size. The learning rate and weight decay values are provided in 7.1. All experiments were conducted on an internal cluster. We used 1500 GPU hours on NVIDIA V100, TitanRTX, and QuadroRTX.

	Backbone Params	Total Params	learning rate	weight decay
ResNet 50	23.5 M	28.1 M	0.0001	0.0001
ViT-small	21.6 M	21.8 M	0.00001	0.0001
SCL	176 k	176 k	0.001	0.0001
WReN	1.5 M	1.5 M	0.0001	0
SCL-ResNet 18	11.2 M	11.6 M	0.0005	0.0001

Table 7.1: **Model sizes and training hyperparameters**.

[1] https://pytorch.org/vision/stable/models.html
[2] https://github.com/facebookresearch/moco-v3
[3] https://github.com/dhh1995/SCL
[4] https://github.com/Fen9/WReN

Model	Score
ResNet 50	12.1
ViT small	2.62
SCL	12.6
WReN	6.76
SCL-ResNet 18	**23.1**

Table 7.2: **Compositionality**: Models are quantitatively evaluated in the curriculum condition. The score is the maximum gain in accuracy across data regimes computed for each task, then averaged across tasks. We observe that the qualitative advantage for SCL-ResNet-18 is consistent with the quantitative evaluation.

A3 Additional Results

We provide more results on compositionality evaluation in Figure A3, Figure A4, Table 7.3, and Table 7.2. The results in the joint training setting are consistent with the individual training setting results. Task difficulty is expanded with more analysis in Figure A5 and Figure A6.

A4 Comparison to SVRT

Synthetic Visual Reasoning Test [Fleuret et al., 2011] (SVRT) is a suite of 23 tasks developed for comparing machines to humans on the semantic description of visual scenes. Each test is a binary classification task based on the rules involved in generating those images. Each image contains randomly generated close contour objects based on a rule such as similarity judgment, spatial reasoning, or numerosity. SVRT tasks were designed such that binary classes cannot be separated based on the appearance of objects, spatial positioning, or any geometric or topological properties of scene components. Figure A8 shows some SVRT examples. CVR takes inspiration from SVRT's scene design—object contours on a white background—and the rules for generating scenes. However, with a set of elementary relations and a method for combining them with a compositionality prior, the 103 tasks proposed in CVR are more diverse than SVRT tasks. CVR also uses the Odd-One-Out task setting, which enables a more general instantiation of rules. For example, task #7 in SVRT requires dissociating images of 3 groups of 2 sim-

ilar shapes from images of 2 groups of 3 similar shapes, as shown in Figure A8. This task is generalized in CVR to a *shape-count* rule where images of n groups of m objects are to be discriminated from images with different counts. In this regard, the odd-one-out task can be considered a 4-shot learning setting for SVRT tasks. Furthermore, CVR is a systematic reorganization of SVRT based on compositionality. It can be used for evaluating generalization, transfer learning, and compositionality, unlike what is attainable with the SVRT.

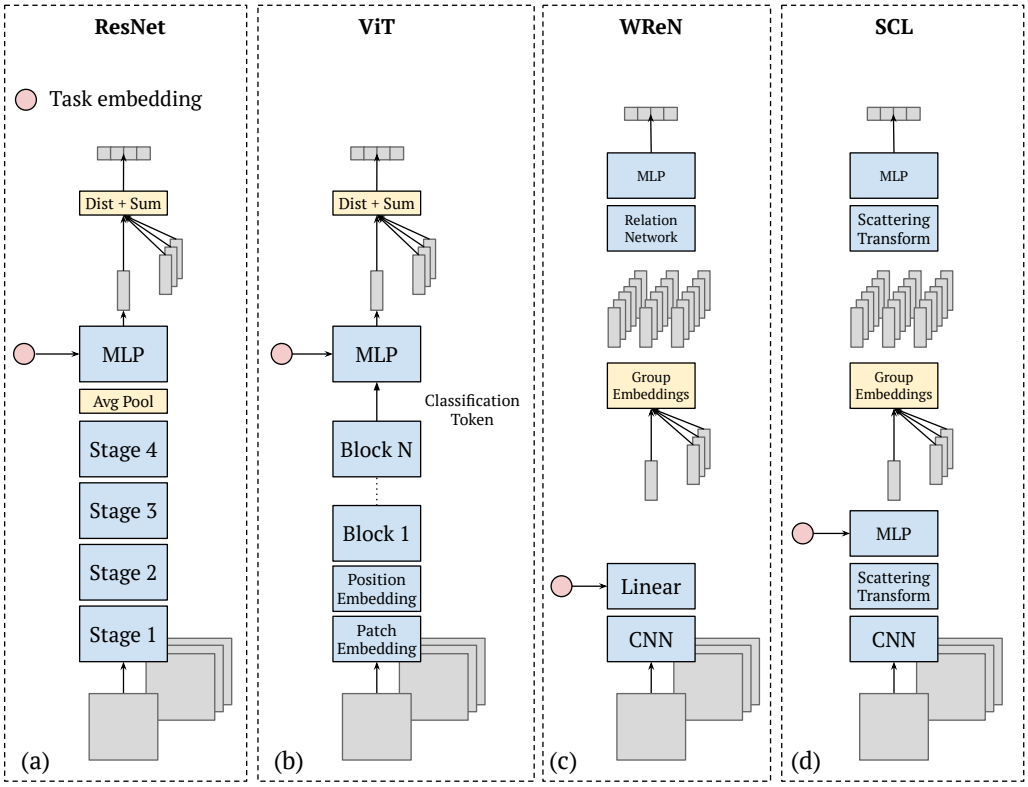
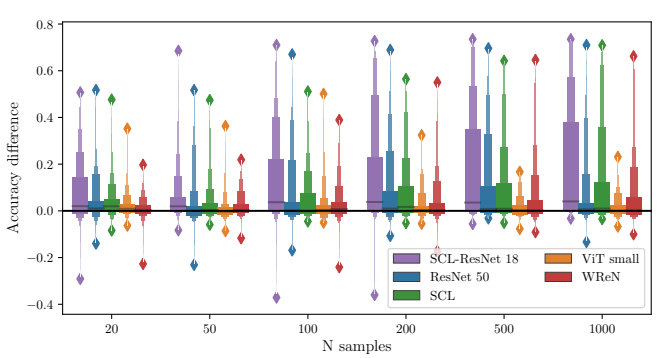
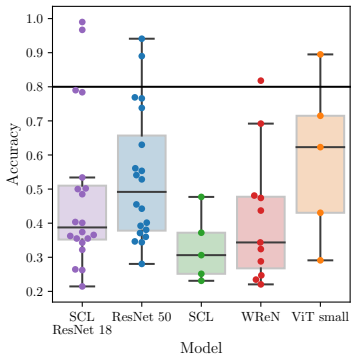


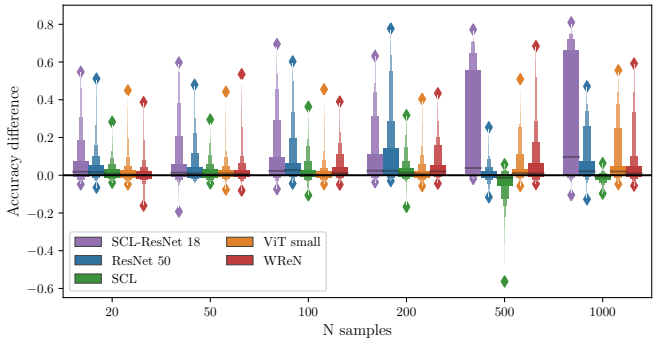
Figure A2: **Model Architectures**: (a) ResNet [He et al., 2015] stages consist of several residual blocks. (b) The patch embedding in transformers splits the image into patches and transforms them into embeddings. Each ViT [Dosovitskiy et al., 2020] block consists of self-attention blocks and MLP transformations; ViT-small uses 12 blocks. (c) WReN [Barrett et al., 2018] is trained in the RPM setting; each image is processed by a CNN, and then all image embeddings are processed by a Relation Network. (d) Similarly to WReN, SCL [Wu et al., 2020] is also trained in the RPM setting. Each image is processed by a CNN and a scattering transformation. All image embeddings are processed by a second scattering transformation. In SCL-ResNet-18, the CNN encoder is substituted with ResNet-18. Details of model architectures can be found in their respective references.



(a) Curriculum - Indivudual setting



(b) Reverse Curriculum



(c) Curriculum - Joint setting

Figure A3: **Compositionality**: We evaluate models' capacity to reuse previous knowledge. **Curriculum**: Models trained with a curriculum are compared to models trained from scratch. The distribution of differences in accuracy across tasks is plotted for each model. **Reverse Curriculum**: In the 1000-sample data regime, we pick rules for which models achieved higher than 80% accuracy, and we evaluate them on the respective elementary rules.

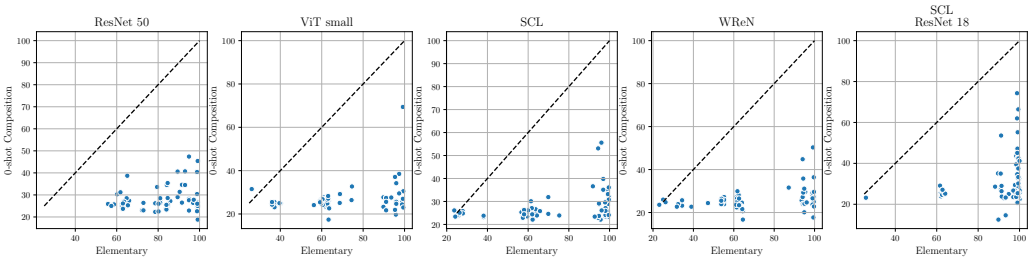


Figure A4: **Compositionality**: Models trained on elementary tasks are zero-shot evaluated on their compositions. Models fail at all compositions without finetuning.

N train samples			20	50	100	200	500	1000
individual	ResNet-50	rand init	25.9	27.7	28.5	29.4	32.6	39.0
		transfer	30.0	30.7	34.2	36.9	42.3	45.0
		difference	4.10	3.06	5.73	7.47	11.5	5.75
	ViT-Small	rand init	25.9	27.0	27.5	28.6	30.4	31.2
		transfer	28.3	27.9	30.0	30.7	31.9	32.9
		difference	2.41	0.89	2.51	2.11	1.44	1.21
	SCL	rand init	26.2	29.3	29.6	29.4	30.6	32.0
		transfer	30.3	32.7	34.9	37.4	40.1	43.0
		difference	4.11	3.43	5.27	7.93	9.49	11.0
	WReN	rand init	28.8	30.5	30.9	31.4	32.4	34.6
		transfer	29.8	32.5	34.0	35.2	37.7	40.4
		difference	1.04	2.03	3.08	3.78	5.28	5.79
	SCL-ResNet-18	rand init	28.7	33.3	32.9	35.5	37.6	41.3
		transfer	36.3	39.1	45.6	49.1	55.1	61.2
		difference	7.60	5.80	12.7	13.6	17.5	20.0
joint	ResNet-50	rand init	25.5	26.2	26.6	29.2	48.6	55.7
		transfer	29.8	30.5	33.8	40.0	49.4	62.9
		difference	4.31	4.30	7.19	10.8	0.86	7.13
	ViT-Small	rand init	25.6	26.0	26.3	26.5	27.0	28.1
		transfer	27.6	27.9	27.9	28.5	31.2	34.9
		difference	2.00	1.97	1.56	1.95	4.14	6.84
	SCL	rand init	25.3	26.2	26.9	27.2	41.8	45.1
		transfer	27.2	28.0	29.0	30.1	37.6	44.0
		difference	1.87	1.90	2.11	2.90	-4.22	-1.07
	WReN	rand init	27.0	26.9	27.7	29.1	33.8	39.1
		transfer	28.0	29.5	31.5	34.2	40.0	44.4
		difference	1.06	2.56	3.86	5.04	6.20	5.21
	SCL-ResNet-18	rand init	26.2	27.5	27.6	30.1	25.8	26.1
		transfer	32.5	34.3	36.9	40.0	48.6	55.5
		difference	6.33	6.76	9.34	9.91	22.7	29.4

Table 7.3: **Curriculum Condition**: Models are pretrained on the elementary tasks before finetuning on the complex tasks (transfer). They are compared to models trained from a random initialization (rand init).

Figure A5: **Task difficulty**: Average accuracy on the elementary rules and their pair-wise compositions. **Individual vs. Joint**: Models are trained on each rule separately or trained jointly on all rules. **Rand-Init vs. SSL**: models are randomly initialized or pretrained with self-supervision.

Figure A6: **Task Difficulty Analysis**: The difference in SES per task is computed in various configurations. **Joint vs. individual rule learning** Results vary over spatial tasks; while some models benefit from joint learning in these tasks (SCL and ResNet50), others have the opposite effects (ViT-small and SCL-ResNet18). **Initializations**: Initializations benefit downstream CVR performance differently. We observe that pretraining improves performance over elementary tasks overall for ResNet50. **Models**: The performance in the joint rule learning setting is compared across models. The comparison shows variations in performance over elementary tasks and spatial tasks.

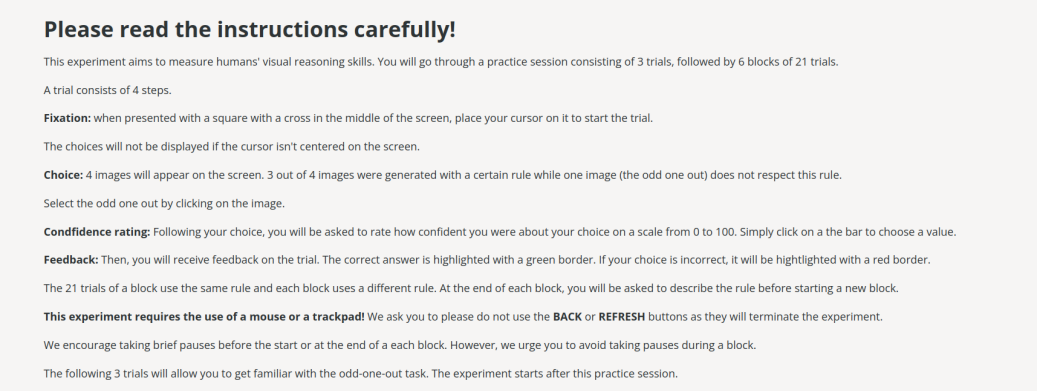


Figure A7: **Behavioral experiment instructions.**

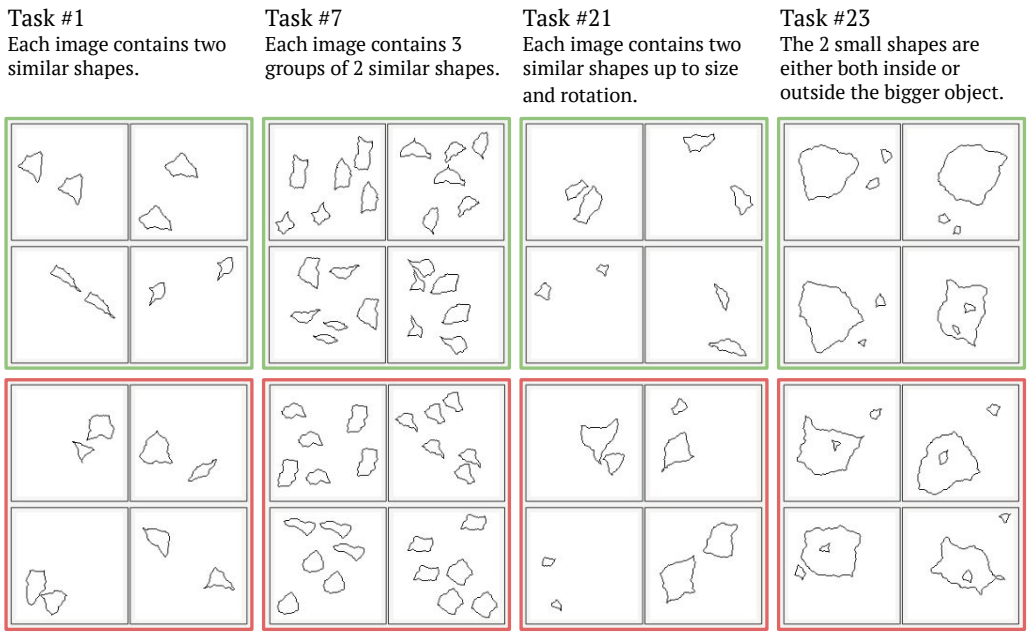
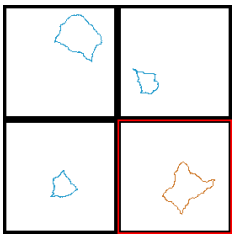
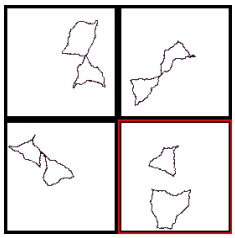


Figure A8: **SVRT task examples**: positive examples are highlighted by a green border and negative examples are highlighted by a red border.

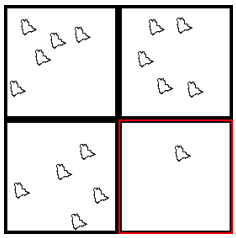
A5 Rule Examples



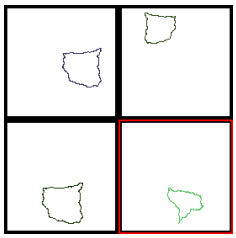
(a) The hue of the object is constant



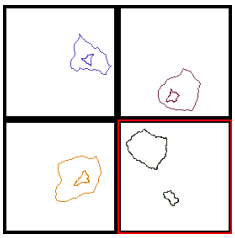
(b) Two objects are in contact



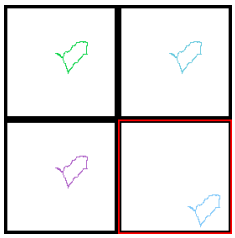
(c) The number of objects is constant



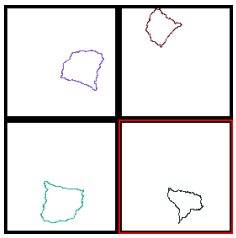
(d) Flips of the same object



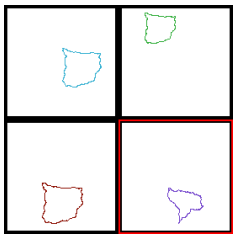
(e) An object contains another object



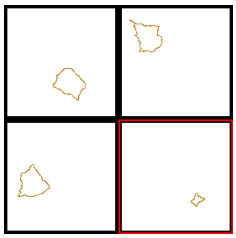
(f) The object is always in the same position



(g) Rotations of the same object

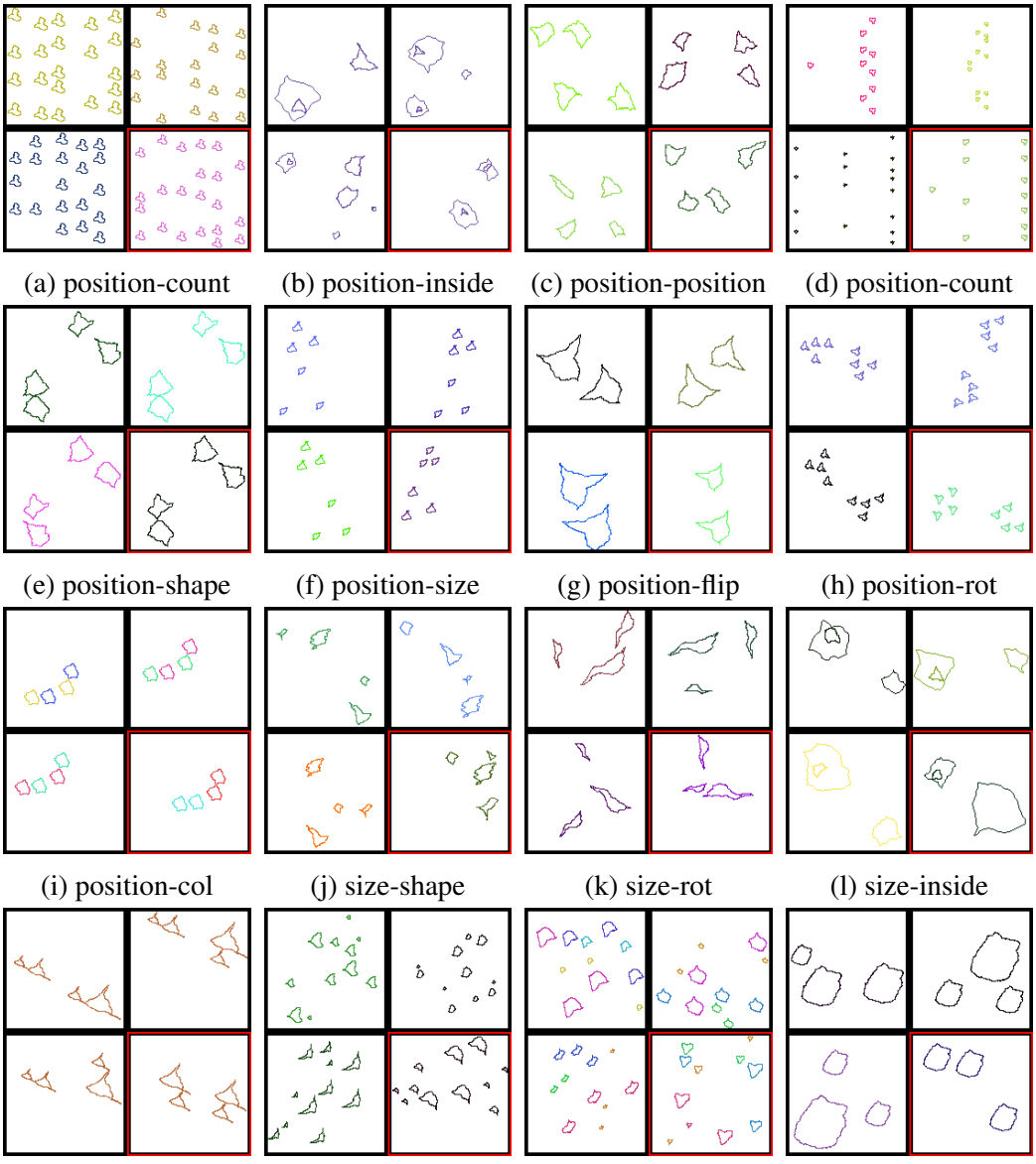


(h) The shape is constant



(i) The size of the object is constant

Figure A9: **Elementary rules**



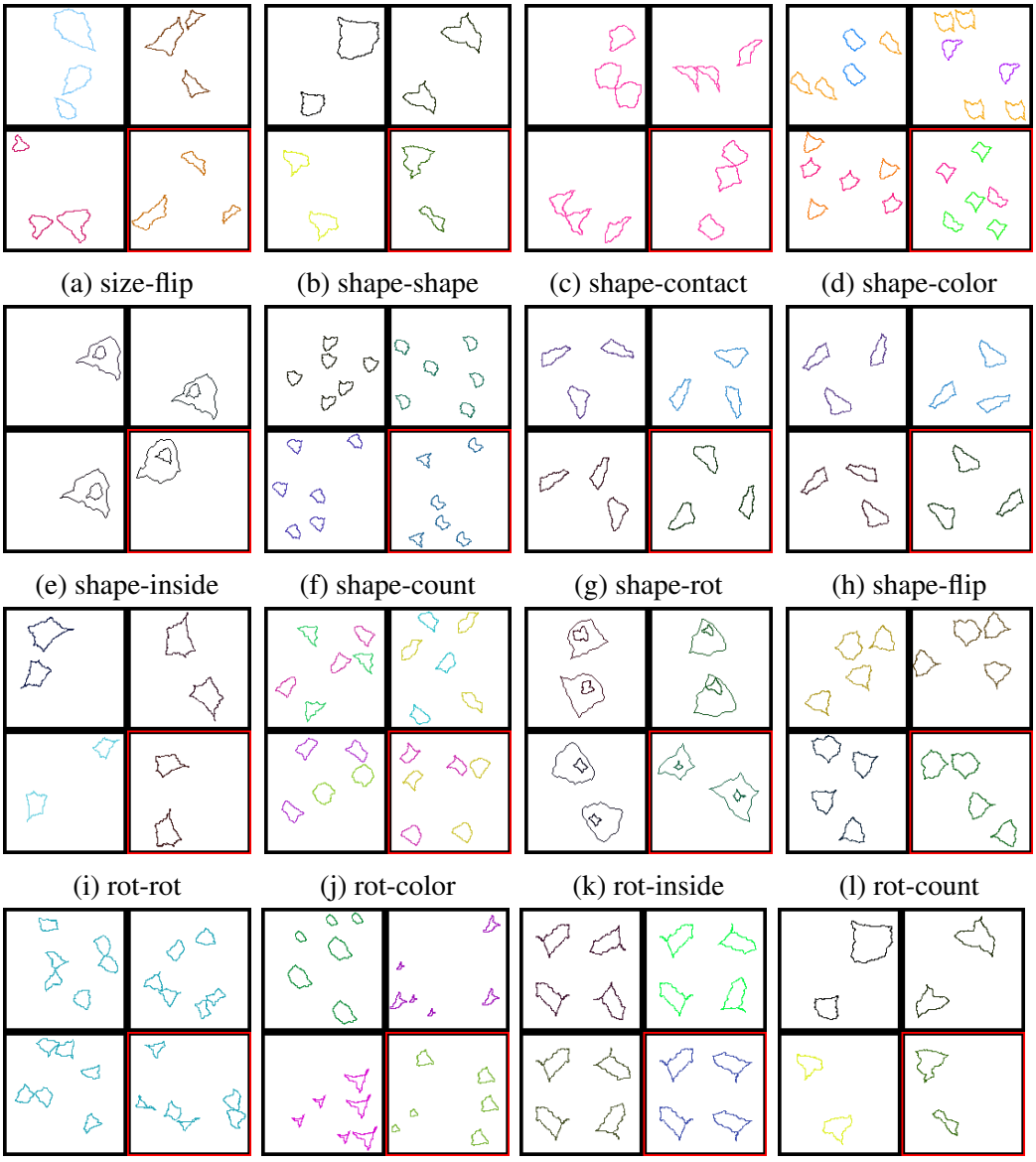
(a) position-count (b) position-inside (c) position-position (d) position-count

(e) position-shape (f) position-size (g) position-flip (h) position-rot

(i) position-col (j) size-shape (k) size-rot (l) size-inside

(m) size-contact (n) size-count (o) size-color (p) size-size

Figure A10: **Composition rules 1**



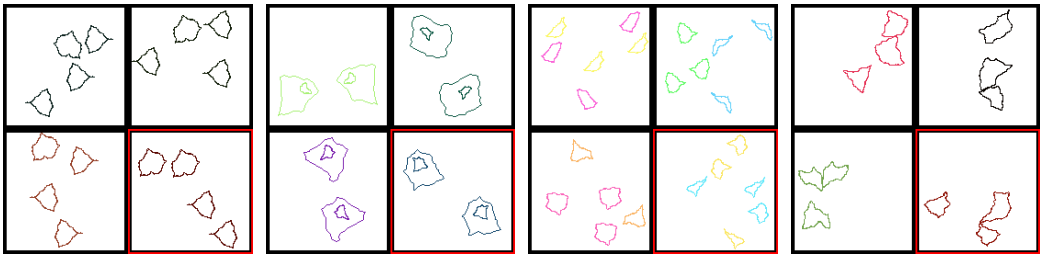
(a) size-flip (b) shape-shape (c) shape-contact (d) shape-color

(e) shape-inside (f) shape-count (g) shape-rot (h) shape-flip

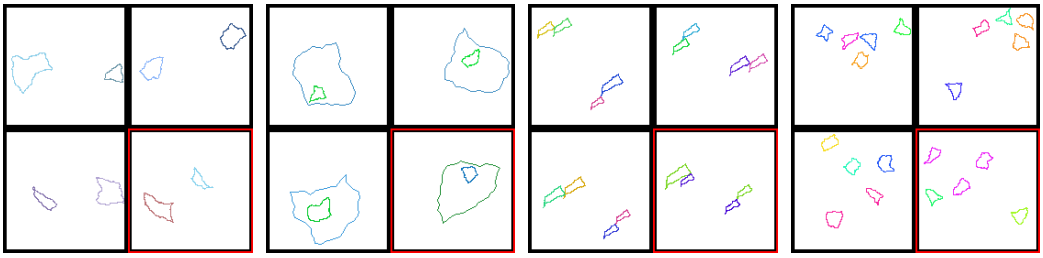
(i) rot-rot (j) rot-color (k) rot-inside (l) rot-count

(m) rot-contact (n) size-count (o) rot-flip (p) flip-flip

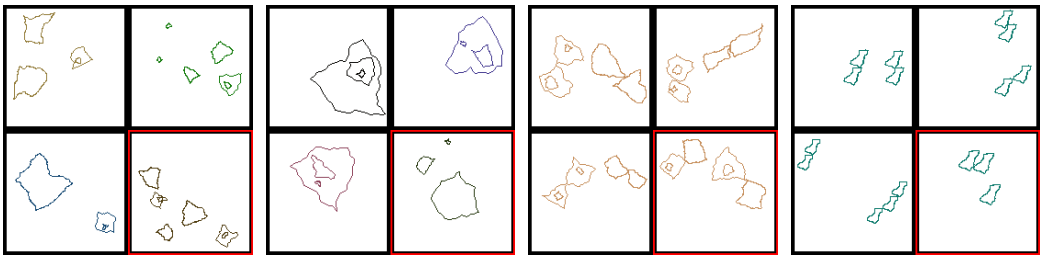
Figure A11: **Composition rules 2**



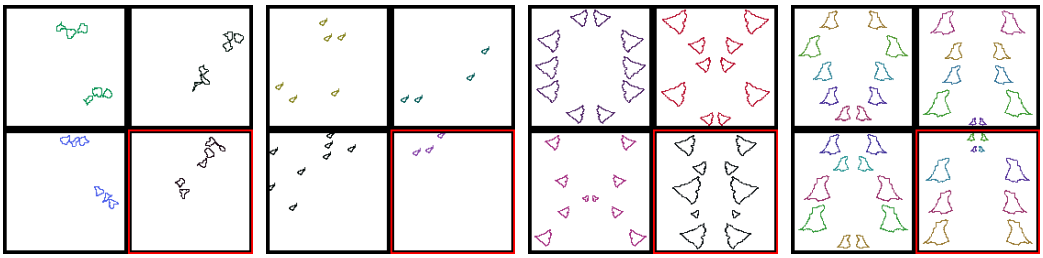
(a) flip-count (b) flip-inside (c) flip-color (d) flip-contact



(e) color-color (f) color-inside (g) color-contact (h) color-count



(i) inside-count (j) inside-inside (k) inside-contact (l) contact-contact



(m) contact-count (n) count-count (o) position-size-flip (p) position-size-color-flip

(q) shape-contact (r) position-inside

Figure A12: **Composition rules 3**

CHAPTER 8

EXPERIMENTS ON COGNITIVE ARCHITECTURES

B1 AbstractNet architecture

The architecture of AbstractNet consists of a set of modules m_i for $i \in \{1, ..., N\}$. Each module has input g and output h gates that can be external g^E (resp. h^E), internal g^I (resp. h^I), or recurrent activity g^r (resp. h^r). Each module is a neural network whose weights are used for processing inputs. given that a module can process inputs in many ways, each module has a set of functions

$$f_i^m(\{g_j^E\}, \{g_k^I\}, \{g_l^R\}) = \{h_m^E\}, \{h_n^I\}, \{h_o^R\}$$

Where are $\{g_j^E\}, \{g_k^I\}), \{g_l^R\}, \{h_m^E\}, \{h_n^I\}, \{h_o^R\}$ are subsets of the module's gates. The controller c module selects which functions f_i^m to use in each module m by sampling from an output vector of action probabilities a_f^t at each time-step t. It also routes information from internal output gates to internal input gates using a routing matrix $R_{i,j}^t$.

$$g_i^{I,t+1} = \sum_j \hat{R}_j^t T(h_j^{I,t}, e_{g_i^I}, e_{h_j^I})$$

where $\hat{R}^t = softmax(R_i^t)$, T is a translation function, and $e_{g_i^I}$, $e_{h_j^I}$ are embeddings for the internal input gate and (resp.) internal output gate. The translation function transforms inputs based on their source (output gate) and target (input gate), it is incorporated to adapt the representational space of the output gate to that of the input gate. It avoids constraining all modules to use the same representational space.

The controller also decides interactions with task instances by sampling its output action probabilities a_I^t which determines if the model reads new task inputs and a_O^t which determines if the model posts outputs to the task instance.

The controller receives as an additional input an embedding that represents decisions and routing from the previous time-step, information about which external input gates received new inputs and which external output gates were used at the previous time-step.

Given one instance of a batch of task instances from various tasks, the model first initializes all vectors and tensors as placeholders for input data, task embedding, routing matrix, action decisions, recurrent states, internal input gates, and external input gates. These tensors constitute the internal state of the model which is updated at each time-step. At each time-step, the controller first decides which functions to use and the routing of latent state contents to the modules specified by the action decisions. After preparing the inputs and executing the functions, the latent state is updated.

Model decisions and task outputs are used for computing the loss at the end of each task instance. AbstractNet is trained end-to-end with task-specific objectives. The weights of modules and networks used for routing are optimized using task-specific losses, while networks used for deciding module activation and task interactions are trained in a reinforcement learning setting. We use advantage actor-critic (A2C) with generalized advantage estimation (GAE).

$$\mathcal{L} = \mathcal{L}_{task} + \alpha \mathcal{L}_{actions}$$

where α is a hyperparameter that weighs the two losses. The reward is chosen as the negative loss computed for the task sample $r = -\mathcal{L}_{task}$. Models with an ablation of the adaptive computation time do not include module activation and task interaction decisions; they are trained with the task-specific loss only.

Additional techniques were used to facilitate and improve the model's train-

ing. Early in training, model decisions on reading external inputs and writing external outputs are overridden. For example, in single output tasks such as image classification, the model is biased to read inputs in the first time-step and give outputs at the last step which corresponds to the maximum number of steps allowed for the model. After a few training steps the constraint is alleviated and the model discovers that it can perform the task in fewer steps. The model penalized for using many internal times steps for processing inputs by introducing a negative reward as a function of the number of steps internal steps.

We attempted other techniques including regularization of softmax temperature used when processing the routing matrix to encourage exploration when learning routing schemes. Similarly, random action selection was introduced for exploration. However, these techniques did not enhance model performance.

The choice of module architectures largely depends on the task. In this work, we used modules of various architectures:

- Gated recurrent unit (GRU) [Cho et al., 2014] as the main architecture for the controller.
- fully internal modules as MLPs and memory modules such as the differentiable neural computer (DNC).
- A 4-layered CNN with ReLU activations and average pooling at the final layer for a vision module.
- A symbol processing module for various types of vector and sequential inputs.
- Embedding modules for embedding text tokens.
- List processing modules that select inputs from a list based on a query.

All models are trained using a fixed set of hyperparameters for all tasks. For AbstractNet, the Adam optimizer is used for updating model weights with a learning rate of 0.0002, no weight decay, and a batch size of 30. The dimension of the input and output vectors of abstract modules is 128.

B2 Task design

Each task has a set of inputs and outputs with their corresponding timestamps (the time-steps at which they are involved in the task), as well as the minimum and maximum numbers of internal computation steps allowed for the model within each task step. The model's architecture is equipped with a controller and a number of MLP internal modules by default. Each task included in the curriculum is accounted for in the architecture with its corresponding external input and output processing modules.

The visual categorization small image datasets MNIST [Deng, 2012] and Cifar10 [Krizhevsky et al., 2014] have one image as an input and a 10-dimensional classification vector as the output. It involves a vision module two classification modules one for each dataset. The loss computed in this task is a cross-entropy loss over the logits provided by the classification modules. The selection task has a list of 16-dimentional vectors as input and a single 16-dimensional vector as the output. It involves a list processing module as the input and an MLP as the output. The cognitive tasks developed by Yang et al. [2019] have a sequence of 32-dimensional vectors as inputs and a sequence of 10-dimensional vectors as the output with their respective timestamps. The model is equipped with an input MLP and an output MLP for processing these task variables. The copy task is similar in inputs and outputs to the cognitive tasks, except the vectors are 16-dimensional. It involves a DNC memory module in addition to the internal MLP modules. The bAbI tasks have a sequence of tokens as input and one token as the output. It involves an embedding module for encoding tokens and decoding embeddings. The task design allows the model to process heterogeneous tasks in the same batch.

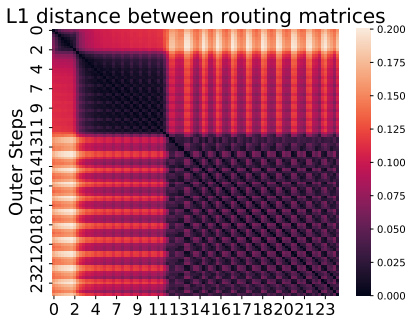
Tasks	AbstractNet		AbstractNet AC		UT		UT ACT	
1 - Single Supporting Fact	99.9	98.8	99.0	97.3	58.6	99.6	47.8	63.6
2 - Two Supporting Facts	35.8	43.7	30.7	38.1	31.1	71.4	30.3	31.0
3 - Three Supporting Facts	24.2	33.1	22.9	31.3	23.9	43.3	29.2	23.9
4 - Two Arg. Relations	92.8	86.3	91.1	96.9	84.4	94.8	85.0	73.1
5 - Three Arg. Relations	80.0	80.6	79.4	78.7	80.7	81.4	81.3	77.3
6 - Yes/No Questions	63.2	89.3	72.2	69.7	52.9	97.8	66.2	70.2
7 - Counting	79.0	77.4	79.1	76.3	76.0	84.3	72.0	72.8
8 - Lists/Sets	86.6	86.2	88.8	85.6	73.5	89.7	71.2	67.0
9 - Simple Negation	70.3	91.4	92.9	70.4	62.3	99.5	60.4	74.9
10 - Indefinite Knowledge	68.1	83.0	86.4	58.5	47.2	96.2	53.9	59.1
11 - Basic Coreference	77.9	99.0	89.5	93.3	70.8	100	67.6	77.3
12 - Conjunction	99.7	99.6	99.8	98.9	79.4	100	69.0	80.6
13 - Compound Coref.	91.7	99.0	94.4	96.3	91.2	99.9	92.5	89.9
14 - Time Reasoning	77.9	77.4	77.3	78.3	37.4	78.6	33.1	36.4
15 - Basic Deduction	68.3	70.2	96.4	51.1	52.3	61.5	51.8	52.8
16 - Basic Induction	45.2	42.9	44.9	43.4	42.5	43.0	44.3	43.5
17 - Positional Reasoning	53.3	55.9	56.1	57.3	57.2	55.1	58.9	57.3
18 - Size Reasoning	92.0	89.6	91.3	91.2	91.9	90.9	92.0	91.5
19 - Path Finding	10.1	09.1	08.8	09.5	08.7	08.0	09.1	09.0
20 - Agent's Motivations	99.8	99.7	100	97.7	97.8	99.4	97.9	98.1
Mean Performance	75.1	76.6	70.8	71.0	60.9	80.3	60.7	62.5

Table 8.1: **Detailed results on bAbi tasks** Single / multi-task performance.

Tasks	AbstractNet		AbstractNet AC		UT		UT ACT	
fdgo	98.57	96.43	98.34	96.72	99.19	97.00	99.44	97.48
reactgo	97.85	95.49	98.05	95.87	98.79	96.66	98.90	96.89
delaygo	97.85	94.64	97.71	96.83	98.61	96.41	98.53	96.85
fdanti	98.72	95.31	98.20	96.42	99.21	97.12	99.08	97.60
reactanti	98.09	95.53	98.33	96.12	99.01	96.66	98.46	97.12
delayanti	97.50	94.30	96.33	96.58	99.12	96.75	98.96	97.11
dm1	89.42	85.78	75.78	85.21	94.80	89.91	93.49	88.86
dm2	90.10	85.05	74.50	84.26	95.09	88.92	94.05	89.02
contextdm1	88.25	82.86	82.58	81.88	94.70	87.67	94.41	86.45
contextdm2	88.53	83.80	76.97	81.36	94.78	88.10	93.85	87.96
multidm	87.21	86.18	87.01	84.49	96.19	89.60	95.66	90.63
delaydm1	98.51	95.91	96.64	96.04	99.01	97.10	99.17	97.24
delaydm2	98.50	96.13	91.65	96.88	98.93	97.22	99.15	97.29
contextdelaydm1	97.97	92.34	95.87	91.70	99.10	97.46	99.25	97.50
contextdelaydm2	97.40	91.67	98.11	92.62	99.07	97.22	99.10	97.24
multidelaydm	97.59	96.04	92.26	96.34	98.93	97.24	99.13	97.11
dmsgo	99.12	97.67	97.59	97.61	99.18	98.00	99.22	98.47
dmsnogo	98.66	95.73	98.31	96.48	95.99	97.12	98.33	97.43
dmcgo	100	98.62	91.10	98.89	100	98.91	100	97.75
dmcnogo	100	99.28	100	99.12	100	99.29	100	98.10
Mean Performance	95.99	92.56	92.26	93.07	97.98	95.81	97.91	95.11

Table 8.2: **Detailed results on cognitive tasks** Single / multi-task performance.

(a) Correlation heatmap



(b) Routing matrix

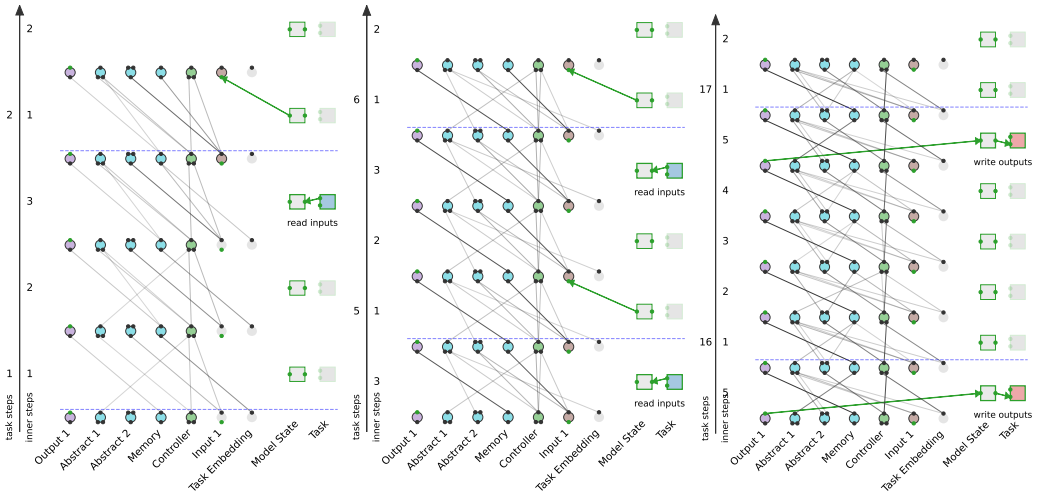
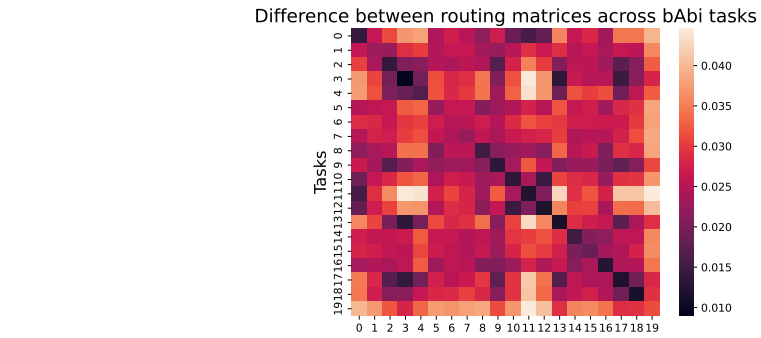


Figure B1: **Routing analysis in the copy task**: a) The heatmap represents the L1 distance between the routing matrices of two successive sets of six computation steps. The two main phases of reading and writing can be distinguished by periods of similar routing matrices: 2–12 and 12–24. b) While certain aspects of routing remain constant, such as routing memory output to the output module, routing between abstract modules highly varies from the reading to the writing phase.

(a) Routing Matrix Analysis.



(b) Task 1 (c) Task 2 (d) Task 4

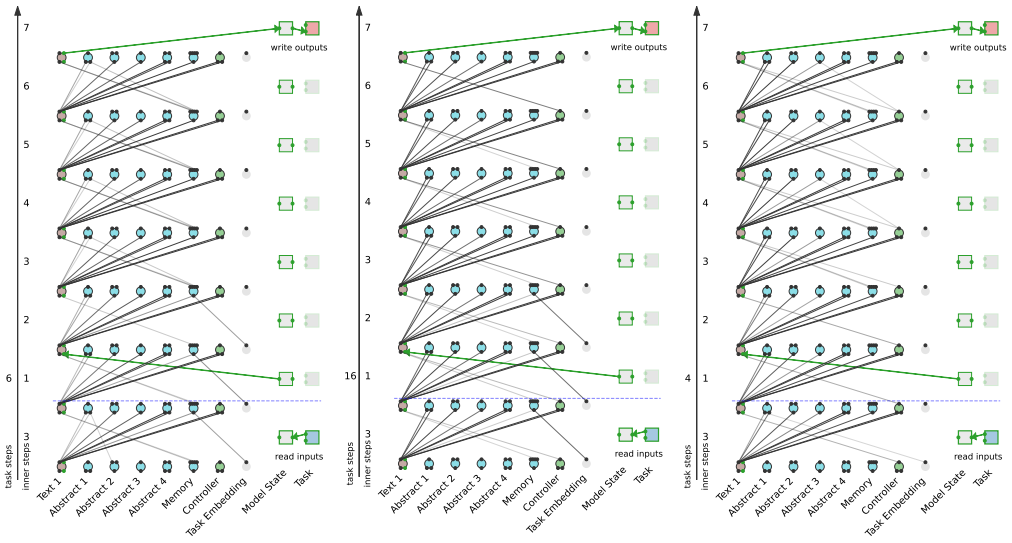


Figure B2: **Routing matrix differences in bAbi tasks**: The routing matrices of the last 7 computation steps are taken from 300 samples of each task; the routing matrices of each sample are compared to others from all tasks, and the differences are averaged within each group. a) Differences are generally small between samples of the same task and vary across tasks. The model uses similar routing strategies for preparing outputs in tasks (3,4,5,14,18) and (1,11,12,13). b) Examples of these routing strategies in tasks 1, 2, and 4, chosen based on accuracy, show the differences in routing information to the output text module; in task 1, the input to the text module is from the memory module, and in task 2, the input is routed from the controller, while in task 4, the input is a combination of controller and memory output.

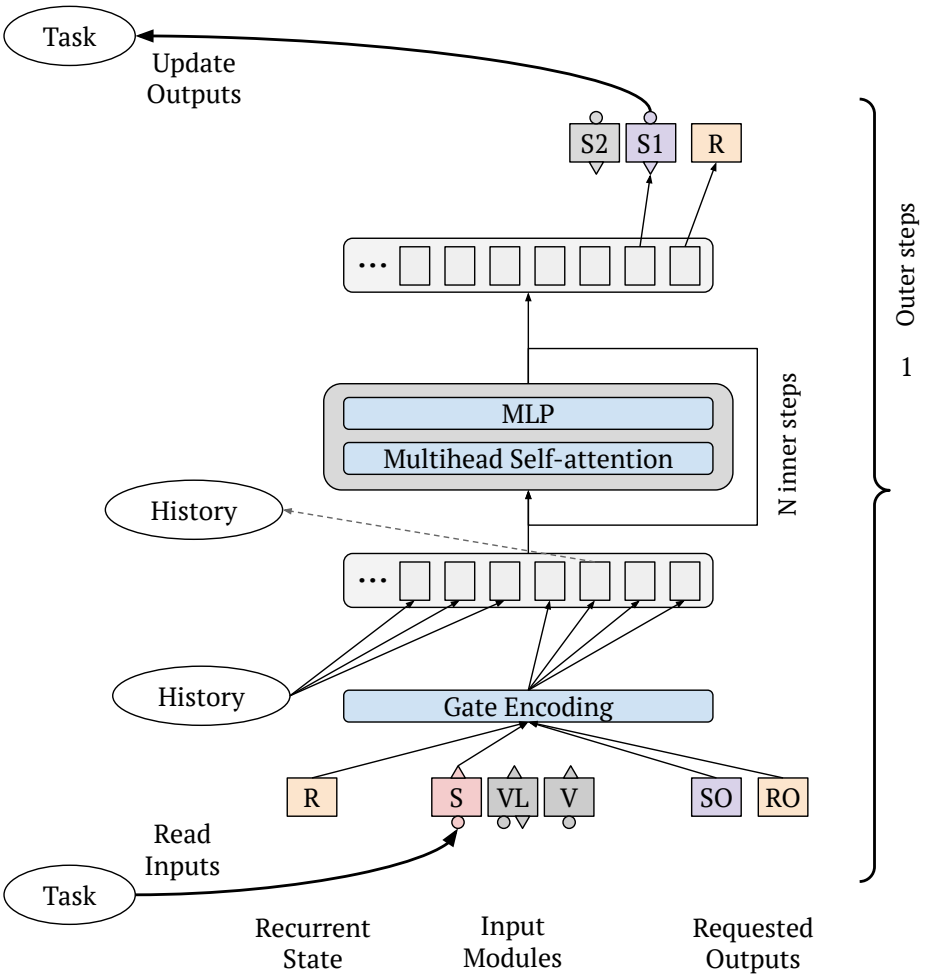


Figure B3: **Adapting Universal Transformer**: The Universal Transforer (UT) consists of a single block of multihead self-attention followed by an MLP that recurrently processes input embeddings. To compare AbstractNet and UT at a similar level, the architecture of UT is augmented with the input and output modules used by AbstractNet. Inner computations are UT-recurrent steps. At each outer step, the model is fed task inputs, a recurrent state embedding, query embeddings used for providing outputs to the task, and a history of input embeddings from previous outer steps. At each outer step, the task inputs are concatenated with the history.

References

BIBLIOGRAPHY

Yuhuai Wu, Honghua Dong, Roger Grosse, and Jimmy Ba. The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *arXiv preprint arXiv:2007.04212*, 2020. (cit. on pp. ix, xii, 24, 37, 38, 114, 126)

Mohit Vaishnav, Remi Cadene, Andrea Alamia, Drew Linsley, Rufin VanRullen, and Thomas Serre. Understanding the computational demands underlying visual reasoning. *Neural Computation*, 34(5):1075–1099, 2022. Publisher: MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info .... (cit. on pp. ix, 24, 37, 38, 42, 114)

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5317–5327, 2019. (cit. on pp. ix, xii, 23, 24, 29, 33, 38, 76, 113, 122)

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International conference on machine learning*, pages 511–520. PMLR, 2018. (cit. on pp. ix, xii, 23, 24, 29, 33, 38, 112, 113, 121, 126)

François Fleuret, Ting Li, Charles Dubout, Emma K Wampler, Steven Yantis, and Donald Geman. Comparing machines and humans on a visual categorization test. *Proc. Natl. Acad. Sci. U. S. A.*, 108(43):17621–17625, October 2011. (cit. on pp. ix, 23, 24, 25, 29, 33, 40, 44, 113, 124)

François Chollet. On the Measure of Intelligence, November 2019. URL http://arxiv.org/abs/1911.01547. arXiv:1911.01547 [cs]. (cit. on pp. ix, 4, 5, 23, 24, 25, 33, 106, 112, 113)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv:1512.03385 [cs]. (cit. on pp. xii, 9, 38, 108, 126)

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Min-

derer, Georg Heigold, Sylvain Gelly, and others. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (cit. on pp. xii, 38, 126)

OpenAI. GPT-4 Technical Report, March 2023. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs]. (cit. on pp. 2, 9, 19, 104, 108, 111)

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4, March 2023. URL http://arxiv.org/abs/2303.12712. arXiv:2303.12712 [cs]. (cit. on pp. 2, 104)

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, April 2022. URL http://arxiv.org/abs/2204.06125. arXiv:2204.06125 [cs]. (cit. on pp. 2, 19, 51, 104, 111, 116)

Gottlob Frege. *The Foundations of Arithmetic: A Logico-Mathematical Enquiry Into the Concept of Number*. Northwestern University Press, December 1980. ISBN 978-0-8101-0605-5. Google-Books-ID: z0KtOtNYMEQC. (cit. on pp. 3, 10, 105)

Jerry A. Fodor. *The Language of Thought*. Harvard University Press, 1975. (cit. on pp. 3, 11, 105)

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building Machines That Learn and Think Like People, November 2016. URL http://arxiv.org/abs/1604.00289. arXiv:1604.00289 [cs, stat]. (cit. on pp. 4, 7, 9, 16, 17, 109)

Shane Legg and Marcus Hutter. A Collection of Definitions of Intelligence, June 2007a. URL http://arxiv.org/abs/0706.3639. arXiv:0706.3639 [cs]. (cit. on pp. 4)

Shane Legg and Marcus Hutter. Universal Intelligence: A Definition of Machine Intelligence, December 2007b. URL http://arxiv.org/abs/0712.3329. arXiv:0712.3329 [cs]. (cit. on pp. 5, 106)

Thomas L. Griffiths. Understanding Human Intelligence through Human Limitations. *Trends in Cognitive Sciences*, 24(11):873–883, November 2020. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2020.09.001. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(20)30215-1. Publisher: Elsevier. (cit. on pp. 7, 8)

Nina Attridge, Andrew Aberdein, and Matthew Inglis. Does studying logic improve logical reasoning? January 2016. URL https://repository.lboro.ac.uk/articles/conference_contribution/Does_studying_logic_improve_logical_reasoning_/9373463/1. Publisher: Loughborough University. (cit. on pp. 8)

Matthew Inglis and Adrian Simpson. MATHEMATICIANS AND THE SELECTION TASK. 2004. (cit. on pp. 8)

Clio Cresswell and Craig Speelman. Does mathematics training lead to better logical thinking and reasoning? A cross-sectional assessment from students to professors. *PLOS ONE*, 15:e0236153, July 2020. doi: 10.1371/journal.pone.0236153. (cit. on pp. 8)

Andrew J. Nam and James L. McClelland. Systematic human learning and generalization from a brief tutorial with explanatory feedback, March 2023. URL http://arxiv.org/abs/2107.06994. arXiv:2107.06994 [cs]. (cit. on pp. 8)

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60 (6):84–90, May 2017. ISSN 0001-0782, 1557-7317. doi: 10.1145/3065386. URL https://dl.acm.org/doi/10.1145/3065386. (cit. on pp. 9, 108)

Sven Eberhardt, Jonah Cader, and Thomas Serre. How Deep is the Feature Analysis underlying Rapid Visual Categorization?, June 2016. URL http://arxiv.org/abs/1606.01167. arXiv:1606.01167 [cs]. (cit. on pp. 9, 108)

Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models

predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, June 2014. doi: 10.1073/pnas.1403112111. URL https://www.pnas.org/doi/10.1073/pnas.1403112111. Publisher: Proceedings of the National Academy of Sciences. (cit. on pp. 9, 108)

Rishi Rajalingham, Kailyn Schmidt, and James J. DiCarlo. Comparison of Object Recognition Behavior in Human and Monkey. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience*, 35(35):12127–12136, September 2015. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.0573-15.2015. (cit. on pp. 9, 108)

Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks, October 2020a. URL http://arxiv.org/abs/1808.08750. arXiv:1808.08750 [cs, q-bio, stat]. (cit. on pp. 9, 76, 108)

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs]. (cit. on pp. 9, 19, 108, 111)

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, February 2023a. URL http://arxiv.org/abs/2302.13971. arXiv:2302.13971 [cs]. (cit. on pp. 9, 19, 108, 111)

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin

Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, July 2023b. URL http://arxiv.org/abs/2307.09288. arXiv:2307.09288 [cs]. (cit. on pp. 9, 19, 108, 111)

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An Embodied Multimodal Language Model, March 2023. URL http://arxiv.org/abs/2303.03378. arXiv:2303.03378 [cs]. (cit. on pp. 9, 19, 108, 111)

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and Applications of Large Language Models, July 2023. URL http://arxiv.org/abs/2307.10169. arXiv:2307.10169 [cs]. (cit. on pp. 9, 109)

Paul Smolensky, Richard Thomas McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. Neurocompositional computing: From the Central Paradox of Cognition to a new generation of AI systems. *AI Magazine*, 43(3):308–322, 2022. ISSN 2371-9621. doi: 10.1002/aaai.12065. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aaai.12065. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/aaai.12065. (cit. on pp. 9, 109)

Barbara H. Partee. Nominal and Temporal Anaphora. *Linguistics and Philosophy*,

7(3):243–286, 1984. doi: 10.1007/bf00627707. Publisher: Springer. (cit. on pp. 10)

Zoltán Gendler Szabó. *The case for compositionality*. Oxford University Press, February 2012. doi: 10.1093/oxfordhb/9780199541072.013.0003. URL https://academic.oup.com/edited-volume/41264/chapter/350861452. (cit. on pp. 10)

Zoltán Gendler Szabó. Compositionality. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2022 edition, 2022. URL https://plato.stanford.edu/archives/fall2022/entries/compositionality/. (cit. on pp. 10)

Steven Pinker. *Language learnability and language development*. Language learnability and language development. Harvard University Press, Cambridge, MA, US, 1984. ISBN 978-0-674-51054-8. Pages: xi, 435. (cit. on pp. 11)

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and Cognitive Architecture: A Critical Analysis. 1988. (cit. on pp. 11, 17, 109)

Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 375(1791):20190307, December 2019. doi: 10.1098/rstb.2019.0307. URL https://royalsocietypublishing.org/doi/10.1098/rstb.2019.0307. Publisher: Royal Society. (cit. on pp. 11)

Gary F. Marcus. *Algebraic Mind: Integrating Connectionism and Cognitive Science*. MIT Press, 2003. ISBN 978-0-262-27908-6. Open Library ID: OL29593517M. (cit. on pp. 11)

Noem Chomsky. *Syntactic structures*. Syntactic structures. Mouton, Oxford, England, 1957. Pages: 116. (cit. on pp. 11)

Stanislas Dehaene, Florent Meyniel, Catherine Wacongne, Liping Wang, and Christophe Pallier. The Neural Representation of Sequences: From Transition Probabilities to Algebraic Patterns and Linguistic Trees. *Neuron*, 88(1):2–19, October 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.09.019. URL https://www.sciencedirect.com/science/article/pii/S089662731500776X. (cit. on pp. 11)

Marc D. Hauser, Noam Chomsky, and W. Tecumseh Fitch. The faculty of language: what is it, who has it, and how did it evolve? *Science (New York, N.Y.)*, 298(5598):1569–1579, November 2002. ISSN 1095-9203. doi: 10.1126/science.298.5598.1569. (cit. on pp. 11)

Michael Rescorla. The Language of Thought Hypothesis. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019. URL https://plato.stanford.edu/archives/sum2019/entries/language-thought/. (cit. on pp. 11)

Steven M. Frankland and Joshua D. Greene. Concepts and Compositionality: In Search of the Brain's Language of Thought. *Annual Review of Psychology*, 71(1):273–303, January 2020. ISSN 0066-4308, 1545-2085. doi: 10.1146/annurev-psych-122216-011829. URL https://www.annualreviews.org/doi/10.1146/annurev-psych-122216-011829. (cit. on pp. 11, 61)

Giosuè Baggio. Compositionality in a Parallel Architecture for Language Processing. *Cognitive Science*, 45(5):e12949, 2021. ISSN 1551-6709. doi: 10.1111/cogs.12949. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12949. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12949. (cit. on pp. 11)

Michael B. Chang, Abhishek Gupta, Sergey Levine, and Thomas L. Griffiths. Automatically Composing Representation Transformations as a Means for Generalization, May 2019. URL http://arxiv.org/abs/1807.04640. arXiv:1807.04640 [cs, stat]. (cit. on pp. 11, 98)

Thomas J. Ringstrom. Reward is not Necessary: How to Create a Compositional Self-Preserving Agent for Life-Long Learning, November 2022. URL http://arxiv.org/abs/2211.10851. arXiv:2211.10851 [cs]. (cit. on pp. 11, 98)

Zeb Kurth-Nelson, Timothy Behrens, Greg Wayne, Kevin Miller, Lennart Luettgau, Ray Dolan, Yunzhe Liu, and Philipp Schwartenbeck. Replay and compositional computation. *Neuron*, 111(4):454–469, February 2023. ISSN 08966273. doi: 10.1016/j.neuron.2022.12.028. URL

https://linkinghub.elsevier.com/retrieve/pii/S0896627322011254. (cit. on pp. 11, 98)

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, December 2015. doi: 10.1126/science.aab3050. URL https://www.science.org/doi/10.1126/science.aab3050. Publisher: American Association for the Advancement of Science. (cit. on pp. 11, 49, 98, 99, 114)

Kevin Ellis, Catherine Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lucas Morales, Luke Hewitt, Armando Solar-Lezama, and Joshua B. Tenenbaum. DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning, June 2020. URL http://arxiv.org/abs/2006.08381. arXiv:2006.08381 [cs]. (cit. on pp. 11, 98)

Philipp Schwartenbeck, Alon Baram, Yunzhe Liu, Shirley Mark, Timothy Muller, Raymond Dolan, Matthew Botvinick, Zeb Kurth-Nelson, and Timothy Behrens. Generative replay for compositional visual understanding in the prefrontal-hippocampal circuit, June 2021. URL https://www.biorxiv.org/content/10.1101/2021.06.06.447249v1. Pages: 2021.06.06.447249 Section: New Results. (cit. on pp. 11)

Daniel C. McNamee, Kimberly L. Stachenfeld, Matthew M. Botvinick, and Samuel J. Gershman. Compositional Sequence Generation in the Entorhinal–Hippocampal System. *Entropy*, 24(12):1791, December 2022. ISSN 1099-4300. doi: 10.3390/e24121791. URL https://www.mdpi.com/1099-4300/24/12/1791. (cit. on pp. 11)

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and Generalization In Emergent Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.407. URL https://www.aclweb.org/anthology/2020.acl-main.407. (cit. on pp. 12)

Eugene Kharitonov and Marco Baroni. Emergent Language Generalization and Acquisition Speed are not tied to Compositionality, April 2020. URL http:

//arxiv.org/abs/2004.03420. arXiv:2004.03420 [cs]. (cit. on pp. 12)

Charlotte Caucheteux, Alexandre Gramfort, and Jean-Remi King. Disentangling Syntax and Semantics in the Brain with Deep Networks, June 2021. URL http://arxiv.org/abs/2103.01620. arXiv:2103.01620 [cs, q-bio]. (cit. on pp. 12)

Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5(1):1–10, February 2022. ISSN 2399-3642. doi: 10.1038/s42003-022-03036-1. URL https://www.nature.com/articles/s42003-022-03036-1. Number: 1 Publisher: Nature Publishing Group. (cit. on pp. 12)

Tai Sing Lee and David Mumford. Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, 20(7):1434–1448, July 2003. ISSN 1084-7529. doi: 10.1364/josaa.20.001434. (cit. on pp. 14)

Alan Yuille and Daniel Kersten. Vision as Bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–308, July 2006. ISSN 1364-6613. doi: 10.1016/j.tics.2006.05.002. URL https://www.sciencedirect.com/science/article/pii/S1364661306001264. (cit. on pp. 14)

Thomas Parr, Noor Sajid, Lancelot Da Costa, M. Berk Mirza, and Karl J. Friston. Generative Models for Active Vision. *Frontiers in Neurorobotics*, 15, 2021. ISSN 1662-5218. URL https://www.frontiersin.org/articles/10.3389/fnbot.2021.651432. (cit. on pp. 14)

Edward C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55:189–208, 1948. ISSN 1939-1471. doi: 10.1037/h0061626. Place: US Publisher: American Psychological Association. (cit. on pp. 15, 62)

Karl Friston, Rosalyn J. Moran, Yukie Nagai, Tadahiro Taniguchi, Hiroaki Gomi, and Josh Tenenbaum. World model learning and inference. *Neural Networks*, 144:573–590, December 2021. ISSN 0893-6080. doi: 10.1016/j.neunet.2021.09.011. URL https://www.sciencedirect.com/science/article/pii/S0893608021003610. (cit. on pp. 15)

Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, November 2013. doi: 10.1073/pnas.1306572110. URL https://www.pnas.org/doi/10.1073/pnas.1306572110. Publisher: Proceedings of the National Academy of Sciences. (cit. on pp. 15, 62)

David Ha and Jürgen Schmidhuber. World Models. March 2018. doi: 10.5281/zenodo.1207631. URL http://arxiv.org/abs/1803.10122. arXiv:1803.10122 [cs, stat]. (cit. on pp. 15)

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives, April 2014. URL http://arxiv.org/abs/1206.5538. arXiv:1206.5538 [cs]. (cit. on pp. 16)

Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-Of-Distribution Generalization: A Survey, August 2021. URL http://arxiv.org/abs/2108.13624. arXiv:2108.13624 [cs]. (cit. on pp. 16)

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, June 2018. URL http://arxiv.org/abs/1711.00350. arXiv:1711.00350 [cs]. (cit. on pp. 17, 18, 34, 109, 110, 111)

Gary Marcus. Deep Learning: A Critical Appraisal, January 2018. URL http://arxiv.org/abs/1801.00631. arXiv:1801.00631 [cs, stat]. (cit. on pp. 17, 109)

Morten H. Christiansen and Nick Chater. Generalization and Connectionist Language Learning. *Mind & Language*, 9(3):273–287, September 1994. ISSN 0268-1064, 1468-0017. doi: 10.1111/j.1468-0017.1994.tb00226.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1468-0017.1994.tb00226.x. (cit. on pp. 17, 109)

Gary F. Marcus. Rethinking Eliminative Connectionism. *Cognitive Psychology*, 37(3):243–282, December 1998. ISSN 0010-0285. doi: 10.1006/cogp.1998.0694. URL https://www.sciencedirect.com/science/article/pii/S0010028598906946. (cit. on pp. 17, 109)

Matthew M. Botvinick and David C. Plaut. Short-term memory for serial order: a recurrent neural network model. *Psychological Review*, 113(2):201–233, April 2006. ISSN 0033-295X. doi: 10.1037/0033-295X.113.2.201. (cit. on pp. 17, 109)

Jeffrey S. Bowers, Markus F. Damian, and Colin J. Davis. A fundamental limitation of the conjunctive codes learned in PDP models of cognition: comment on Botvinick and Plaut (2006). *Psychological Review*, 116(4):986–997, October 2009. ISSN 0033-295X. doi: 10.1037/a0017097. (cit. on pp. 17, 109)

Matthew M. Botvinick and David C. Plaut. Empirical and computational support for context-dependent representations of serial order: reply to Bowers, Damian, and Davis (2009). *Psychological Review*, 116(4):998–1002, October 2009. ISSN 0033-295X. doi: 10.1037/a0017113. (cit. on pp. 17, 62, 109)

Stefan L. Frank, Willem F. G. Haselager, and Iris van Rooij. Connectionist semantic systematicity. *Cognition*, 110(3):358–379, March 2009. ISSN 0010-0277. doi: 10.1016/j.cognition.2008.11.013. URL https://www.sciencedirect.com/science/article/pii/S0010027708002837. (cit. on pp. 17, 109)

Samuel R. Bowman, Christopher D. Manning, and Christopher Potts. Tree-structured composition in neural networks without tree-structured architectures, November 2015. URL http://arxiv.org/abs/1506.04834. arXiv:1506.04834 [cs]. (cit. on pp. 17, 34, 109)

Stefan L. Frank. Getting Real about Systematicity. In Paco Calvo and John Symons, editors, *The Architecture of Cognition: Rethinking Fodor and Pylyshyn's Systematicity Challenge*, page 0. The MIT Press, May 2014. ISBN 978-0-262-02723-6. doi: 10.7551/mitpress/9780262027236.003.0006. URL https://doi.org/10.7551/mitpress/9780262027236.003.0006. (cit. on pp. 17, 109)

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216, November 1990. ISSN 0004-3702. doi: 10.1016/0004-3702(90)90007-M. URL https://www.sciencedirect.com/science/article/pii/000437029090007M. (cit. on pp. 17, 109)

D. D. Hoffman and W. A. Richards. Parts of recognition. *Cognition*, 18(1): 65–96, December 1984. ISSN 0010-0277. doi: 10.1016/0010-0277(84)90022-2. URL https://www.sciencedirect.com/science/article/pii/0010027784900222. (cit. on pp. 17, 109)

Irving Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32(1):29–73, October 1985. ISSN 0734-189X. doi: 10.1016/0734-189X(85)90002-7. URL https://www.sciencedirect.com/science/article/pii/0734189X85900027. (cit. on pp. 17, 109)

Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks, November 2013. URL http://arxiv.org/abs/1311.2901. arXiv:1311.2901 [cs]. (cit. on pp. 17, 110)

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170. (cit. on pp. 17, 110)

Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning Unseen Concepts via Hierarchical Decomposition and Composition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10245–10253, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.01026. URL https://ieeexplore.ieee.org/document/9156655/. (cit. on pp. 18, 110)

Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open World Compositional Zero-Shot Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5218–5226, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00518. URL https://ieeexplore.ieee.org/document/9578210/. (cit. on pp. 18, 110)

Ishan Misra, Abhinav Gupta, and Martial Hebert. From Red Wine to Red Tomato: Composition with Context. In *2017 IEEE Conference on Computer Vi-*

sion and Pattern Recognition (CVPR), pages 1160–1169, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.129. URL http://ieeexplore.ieee.org/document/8099612/. (cit. on pp. 18, 110)

Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning Graph Embeddings for Compositional Zero-shot Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 953–962, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00101. URL https://ieeexplore.ieee.org/document/9577736/. (cit. on pp. 18, 110)

Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-Driven Modular Networks for Zero-Shot Compositional Learning, May 2019. URL http://arxiv.org/abs/1905.05908. arXiv:1905.05908 [cs]. (cit. on pp. 18, 110)

Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition, November 2020. URL http://arxiv.org/abs/2006.14610. arXiv:2006.14610 [cs]. (cit. on pp. 18, 110)

Xin Wang, Fisher Yu, Trevor Darrell, and Joseph E. Gonzalez. Task-Aware Feature Generation for Zero-Shot Compositional Learning, March 2020. URL http://arxiv.org/abs/1906.04854. arXiv:1906.04854 [cs]. (cit. on pp. 18, 110)

Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning Shape Abstractions by Assembling Volumetric Primitives, August 2018. URL http://arxiv.org/abs/1612.00404. arXiv:1612.00404 [cs]. (cit. on pp. 18, 110)

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Inferring and Executing Programs for Visual Reasoning, May 2017a. URL http://arxiv.org/abs/1705.03633. arXiv:1705.03633 [cs]. (cit. on pp. 18, 19, 51, 110, 111, 115)

Izzeddin Gur, Natasha Jaques, Yingjie Miao, Jongwook Choi, Manoj Tiwari, Honglak Lee, and Aleksandra Faust. Environment Generation for Zero-Shot Compositional Reinforcement Learning, January 2022. URL http:

//arxiv.org/abs/2201.08896. arXiv:2201.08896 [cs]. (cit. on pp. 18, 110)

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. MEASURING COMPOSITIONAL GENERALIZATION: A COMPREHENSIVE METHOD ON REALISTIC DATA. 2020. (cit. on pp. 18, 110)

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing Mathematical Reasoning Abilities of Neural Models, April 2019. URL http://arxiv.org/abs/1904.01557. arXiv:1904.01557 [cs, stat]. (cit. on pp. 18, 34, 110)

João Loula, Marco Baroni, and Brenden Lake. Rearranging the Familiar: Testing Compositional Generalization in Recurrent Networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5413. URL https://aclanthology.org/W18-5413. (cit. on pp. 18, 111)

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures, September 2021. URL http://arxiv.org/abs/2007.08970. arXiv:2007.08970 [cs]. (cit. on pp. 18, 111)

Brenden M. Lake and Steven T. Piantadosi. People infer recursive visual concepts from just a few examples, July 2019. URL http://arxiv.org/abs/1904.08034. arXiv:1904.08034 [cs, q-bio, stat]. (cit. on pp. 18, 111)

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: how do neural networks generalise?, February 2020. URL http://arxiv.org/abs/1908.08351. arXiv:1908.08351 [cs, stat]. (cit. on pp. 18, 34, 111)

Joris Baan, Jana Leible, Mitja Nikolaus, David Rau, Dennis Ulmer, Tim Baumgärtner, Dieuwke Hupkes, and Elia Bruni. On the Realization of Compositionality in Neural Networks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*,

pages 127–137, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4814. URL https://www.aclweb.org/anthology/W19-4814. (cit. on pp. 18, 111)

Dieuwke Hupkes, Anand Singh, Kris Korrel, German Kruszewski, and Elia Bruni. Learning compositionally through attentive guidance, July 2019. URL http://arxiv.org/abs/1805.09657. arXiv:1805.09657 [cs]. (cit. on pp. 18, 111)

Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent, February 2020. URL http://arxiv.org/abs/1910.00571. arXiv:1910.00571 [cs]. (cit. on pp. 18, 111)

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to Compose Neural Networks for Question Answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1545–1554, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1181. URL https://aclanthology.org/N16-1181. (cit. on pp. 19, 74, 111)

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to Reason: End-to-End Module Networks for Visual Question Answering, September 2017. URL http://arxiv.org/abs/1704.05526. arXiv:1704.05526 [cs]. (cit. on pp. 19, 111)

Jake Russin, Jason Jo, Randall C. O'Reilly, and Yoshua Bengio. Compositional generalization in a deep seq2seq model by separating syntax and semantics, May 2019. URL http://arxiv.org/abs/1904.09708. arXiv:1904.09708 [cs, stat]. (cit. on pp. 19, 111)

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, April 2022. URL http://arxiv.org/abs/2112.10752. arXiv:2112.10752 [cs]. (cit. on pp. 19, 111)

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation

Models, June 2022. URL http://arxiv.org/abs/2205.01917. arXiv:2205.01917 [cs]. (cit. on pp. 19, 51, 111, 116)

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and Fate: Limits of Transformers on Compositionality, June 2023. URL http://arxiv.org/abs/2305.18654. arXiv:2305.18654 [cs]. (cit. on pp. 19, 112)

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2 (11):665–673, November 2020b. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://www.nature.com/articles/s42256-020-00257-z. Number: 11 Publisher: Nature Publishing Group. (cit. on pp. 23)

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. (cit. on pp. 23)

Shimon Ullman. Visual routines. In *Readings in computer vision*, pages 298–328. Elsevier, 1987. (cit. on pp. 25, 34, 35, 98, 113)

Jacek Mańdziuk and Adam Żychowski. Deepiq: A human-inspired ai system for solving iq test problems. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. (cit. on pp. 26)

Junkyung Kim, Matthew Ricci, and Thomas Serre. Not-So-CLEVR: learning same–different relations strains feedforward neural networks. *Interface focus*, 8(4):20180011, 2018. Publisher: The Royal Society. (cit. on pp. 29)

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017b. (cit. on pp. 33, 34)

Zechen Li and Anders Søgaard. QLEVR: A Diagnostic Dataset for Quantificational Language and Elementary Visual Reasoning. *arXiv preprint arXiv:2205.03075*, 2022. (cit. on pp. 33)

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019. (cit. on pp. 33)

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. *Advances in Neural Information Processing Systems*, 32, 2019. (cit. on pp. 33)

Henry R Burke. Raven's Progressive Matrices (1938): More on norms, reliability, and validity. *Journal of Clinical Psychology*, 41(2):231–235, 1985. Publisher: Wiley Online Library. (cit. on pp. 33)

Ke Wang and Zhendong Su. Automatic generation of raven's progressive matrices. In *Twenty-fourth international joint conference on artificial intelligence*, 2015. (cit. on pp. 33)

Weili Nie, Zhiding Yu, Lei Mao, Ankit B Patel, Yuke Zhu, and Anima Anandkumar. Bongard-LOGO: A new benchmark for human-level concept learning and reasoning. *Adv. Neural Inf. Process. Syst.*, 33:16468–16480, 2020. (cit. on pp. 33)

Mikhail Moiseevich Bongard. The recognition problem. Technical report, FOREIGN TECHNOLOGY DIV WRIGHT-PATTERSON AFB OHIO, 1968. (cit. on pp. 33)

Damien Teney, Peng Wang, Jiewei Cao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. V-prom: A benchmark for visual reasoning using visual progressive matrices. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12071–12078, 2020. Issue: 07. (cit. on pp. 33)

Huaizu Jiang, Xiaojian Ma, Weili Nie, Zhiding Yu, Yuke Zhu, and Anima Anandkumar. Bongard-HOI: Benchmarking Few-Shot Visual Reasoning for Human-Object Interactions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. (cit. on pp. 33)

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016. Publisher: MIT Press. (cit. on pp. 34)

Laura Ruis, Jacob Andreas, Marco Baroni, Diane Bouchacourt, and Brenden M Lake. A benchmark for systematic generalization in grounded language understanding. *Advances in neural information processing systems*, 33:19861–19872, 2020. (cit. on pp. 34)

Zhengxuan Wu, Elisa Kreiss, Desmond C Ong, and Christopher Potts. ReaSCAN: Compositional reasoning in language grounding. *arXiv preprint arXiv:2109.08994*, 2021. (cit. on pp. 34)

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022. (cit. on pp. 34)

Aishwarya Agrawal, Aniruddha Kembhavi, Dhruv Batra, and Devi Parikh. C-vqa: A compositional split of the visual question answering (vqa) v1. 0 dataset. *arXiv preprint arXiv:1704.08243*, 2017. (cit. on pp. 34)

Patrick Cavanagh. Visual cognition. *Vision research*, 51(13):1538–1551, 2011. Publisher: Elsevier. (cit. on pp. 34)

Junkyung Kim, Drew Linsley, Kalpit Thakkar, and Thomas Serre. Disentangling neural mechanisms for perceptual grouping. *arXiv preprint arXiv:1906.01558*, 2019. (cit. on pp. 34, 35, 101)

Drew Linsley, Junkyung Kim, Alekh Ashok, and Thomas Serre. Recurrent neural circuits for contour detection. *arXiv preprint arXiv:2010.15314*, 2020. (cit. on pp. 34)

Guillermo Puebla and Jeffrey S Bowers. Can deep convolutional neural networks support relational reasoning in the same-different task? Publication Title: bioRxiv, September 2021. (cit. on pp. 34)

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. *Advances in neural information processing systems*, 31, 2018. (cit. on pp. 35, 101)

Nicola Messina, Giuseppe Amato, Fabio Carrara, Claudio Gennaro, and Fabrizio Falchi. Recurrent Vision Transformer for Solving Visual Reasoning Problems. *arXiv preprint arXiv:2111.14576*, 2021. (cit. on pp. 38)

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017. (cit. on pp. 38, 51, 115)

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021a. (cit. on pp. 39, 123)

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. (cit. on pp. 39, 121)

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, July 2021. URL https://proceedings.mlr.press/v139/radford21a.html. (cit. on pp. 39)

Ronald B. Dekker, Fabian Otto, and Christopher Summerfield. Curriculum learning for human compositional generalization. *Proceedings of the National Academy of Sciences*, 119(41):e2205582119, October 2022. doi: 10.1073/pnas.2205582119. URL https://www.pnas.org/doi/10.1073/pnas.2205582119. Publisher: Proceedings of the National Academy of Sciences. (cit. on pp. 50, 101, 115)

Tina Chen, Renran Tian, and Zhengming Ding. Visual reasoning using graph convolutional networks for predicting pedestrian crossing intention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3103–3109, 2021b. (cit. on pp. 51, 115)

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016b. (cit. on pp. 51, 65, 115)

Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 655–664, 2021c. (cit. on pp. 51, 115)

Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018. (cit. on pp. 51, 115)

Drew A. Hudson and Christopher D. Manning. Learning by Abstraction: The Neural State Machine, November 2019. URL http://arxiv.org/abs/1907.03950. arXiv:1907.03950 [cs]. (cit. on pp. 51, 74, 115)

Sarthak Mittal, Sharath Chandra Raparthy, Irina Rish, Yoshua Bengio, and Guillaume Lajoie. Compositional attention: Disentangling search and retrieval. *arXiv preprint arXiv:2110.09419*, 2021. (cit. on pp. 51, 115)

Nasim Rahaman, Muhammad Waleed Gondal, Shruti Joshi, Peter Gehler, Yoshua Bengio, Francesco Locatello, and Bernhard Schölkopf. Dynamic Inference with Neural Interpreters, October 2021. URL http://arxiv.org/abs/2110.06399. arXiv:2110.06399 [cs]. (cit. on pp. 51, 115)

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019. (cit. on pp. 51, 65, 69, 115)

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. (cit. on pp. 51, 116)

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. (cit. on pp. 51, 116)

Nanbo Li, Cian Eastwood, and Robert Fisher. Learning object-centric representations of multi-object scenes from multiple views. *Advances in Neural Information Processing Systems*, 33:5656–5666, 2020. (cit. on pp. 51, 116)

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. Attention over learned object embeddings enables complex visual reasoning. *Advances in neural information processing systems*, 34:9112–9124, 2021. (cit. on pp. 51, 116)

H. Burton. Visual Cortex Activity in Early and Late Blind People. *The Journal of Neuroscience*, 23(10):4005–4011, May 2003. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.23-10-04005.2003. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3667661/. (cit. on pp. 57)

Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, 1988. (cit. on pp. 57)

Rufin VanRullen and Ryota Kanai. Deep learning and the Global Workspace Theory. *Trends in Neurosciences*, 44(9):692–704, September 2021. ISSN 0166-2236, 1878-108X. doi: 10.1016/j.tins.2021.04.005. URL https://www.cell.com/trends/neurosciences/abstract/S0166-2236(21)00077-1. Publisher: Elsevier. (cit. on pp. 57)

Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination Among Neural Modules Through a Shared Global Workspace, March 2022. URL http://arxiv.org/abs/2103.01197. arXiv:2103.01197 [cs, stat]. (cit. on pp. 57, 67, 68, 69, 71)

Hisham E Atallah, Michael J Frank, and Randall C O'Reilly. Hippocampus, cortex, and basal ganglia: Insights from computational models of complementary learning systems. *Neurobiology of learning and memory*, 82(3):253–267, 2004. (cit. on pp. 58)

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, July 1995. ISSN 0033-295X. doi: 10.1037/0033-295X.102.3.419. (cit. on pp. 58)

Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. *Trends in Cognitive Sciences*, 20(7):512–534, July 2016. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2016.05.004. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(16)30043-2. Publisher: Elsevier. (cit. on pp. 58)

Matthew M. Botvinick and Jonathan D. Cohen. The Computational and Neural Basis of Cognitive Control: Charted Territory and New Frontiers. *Cognitive Science*, 38(6):1249–1285, 2014. ISSN 1551-6709. doi: 10.1111/cogs.12126. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12126. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cogs.12126. (cit. on pp. 58, 60)

J Ridley Stroop. Studies of interference in serial verbal reactions. *Journal of experimental psychology*, 18(6):643, 1935. (cit. on pp. 59)

Jonathan D Cohen, Kevin Dunbar, and James L McClelland. On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological review*, 97(3):332, 1990. (cit. on pp. 59, 60)

Richard Cooper and Tim Shallice. Contention scheduling and the control of routine activities. *Cognitive neuropsychology*, 17(4):297–338, 2000. (cit. on pp. 59)

Peter Dayan. Bilinearity, rules, and prefrontal cortex. *Frontiers in computational neuroscience*, 1:73, 2007. (cit. on pp. 59)

Stanislas Dehaene and Jean-Pierre Changeux. A hierarchical neuronal network for planning behavior. *Proceedings of the National Academy of Sciences*, 94 (24):13293–13298, 1997. (cit. on pp. 59)

Amitai Shenhav, Matthew M Botvinick, and Jonathan D Cohen. The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron*, 79(2):217–240, 2013. (cit. on pp. 59)

Sebastian Musslick. *On the Rational Bounds of Cognitive Control*. PhD thesis, 2021. (cit. on pp. 59)

Nicolas P. Rougier, David C. Noelle, Todd S. Braver, Jonathan D. Cohen, and Randall C. O'Reilly. Prefrontal cortex and flexible cognitive control: Rules without symbols. *Proceedings of the National Academy of Sciences*, 102 (20):7338–7343, May 2005. doi: 10.1073/pnas.0502455102. URL https://www.pnas.org/doi/10.1073/pnas.0502455102. Publisher: Proceedings of the National Academy of Sciences. (cit. on pp. 60)

Anne GE Collins and Michael J Frank. Cognitive control over learning: creating, clustering, and generalizing task-set structure. *Psychological review*, 120(1): 190, 2013. (cit. on pp. 60)

Randall C O'Reilly and Michael J Frank. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural computation*, 18(2):283–328, 2006a. (cit. on pp. 60)

Gary Aston-Jones and Jonathan D Cohen. An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance. *Annu. Rev. Neurosci.*, 28:403–450, 2005. (cit. on pp. 60)

Todd S Braver and Jonathan D Cohen. On the control of control: The role of dopamine in regulating prefrontal function and working memory. *Control of cognitive processes: Attention and performance XVIII*, (2000), 2000. (cit. on pp. 60)

Earl K Miller and Jonathan D Cohen. An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, 24(1):167–202, 2001a. (cit. on pp. 60)

John Duncan. The multiple-demand (md) system of the primate brain: mental programs for intelligent behaviour. *Trends in cognitive sciences*, 14(4):172–179, 2010. (cit. on pp. 60)

John Duncan and Adrian M Owen. Common regions of the human frontal lobe recruited by diverse cognitive demands. *Trends in neurosciences*, 23(10): 475–483, 2000. (cit. on pp. 60)

E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24:167–202, 2001b. ISSN 0147-006X. doi: 10.1146/annurev.neuro.24.1.167. (cit. on pp. 60)

John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036, 2004. (cit. on pp. 60)

Etienne Koechlin and Christopher Summerfield. An information theoretical approach to prefrontal executive function. *Trends in cognitive sciences*, 11(6): 229–235, 2007. (cit. on pp. 60)

Nico U. F. Dosenbach, Damien A. Fair, Alexander L. Cohen, Bradley L. Schlaggar, and Steven E. Petersen. A dual-networks architecture of top-down control. *Trends in Cognitive Sciences*, 12(3):99–105, March 2008. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2008.01.001. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(08)00027-2. Publisher: Elsevier. (cit. on pp. 60)

Vinod Menon and Lucina Q Uddin. Saliency, switching, attention and control: a network model of insula function. *Brain structure and function*, 214:655–667, 2010. (cit. on pp. 60)

Dagmar Zeithamova, Michael L. Mack, Kurt Braunlich, Tyler Davis, Carol A. Seger, Marlieke T. R. van Kesteren, and Andreas Wutz. Brain Mechanisms of Concept Learning. *Journal of Neuroscience*, 39(42):8259–8266, October 2019. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.1166-19.2019. URL https://www.jneurosci.org/content/39/42/8259. Publisher: Society for Neuroscience Section: Symposium and Mini-Symposium. (cit. on pp. 61)

Michael L Mack, Bradley C Love, and Alison R Preston. Dynamic updating of hippocampal object representations reflects new conceptual knowledge. *Proceedings of the National Academy of Sciences*, 113(46):13203–13208, 2016. (cit. on pp. 61)

Alexandra O Constantinescu, Jill X O'Reilly, and Timothy EJ Behrens. Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352 (6292):1464–1468, 2016. (cit. on pp. 61)

Michael L Mack, Alison R Preston, and Bradley C Love. Ventromedial prefrontal cortex compression during concept learning. *Nature communications*, 11(1): 46, 2020. (cit. on pp. 61)

Kalina Christoff and John DE Gabrieli. The frontopolar cortex and human cognition: Evidence for a rostrocaudal hierarchical organization within the human prefrontal cortex. *Psychobiology*, 28(2):168–186, 2000. (cit. on pp. 61)

Kalina Christoff, Kamyar Keramatian, Alan M Gordon, Rachelle Smith, and Burkhard Mädler. Prefrontal organization of cognitive control according to levels of abstraction. *Brain research*, 1286:94–105, 2009. (cit. on pp. 61)

Silvia A Bunge, Itamar Kahn, Jonathan D Wallis, Earl K Miller, and Anthony D Wagner. Neural circuits subserving the retrieval and maintenance of abstract rules. *Journal of neurophysiology*, 90(5):3419–3428, 2003. (cit. on pp. 61)

Kalina Christoff, Vivek Prabhakaran, Jennifer Dorfman, Zuo Zhao, James K Kroger, Keith J Holyoak, and John DE Gabrieli. Rostrolateral prefrontal cortex involvement in relational integration during reasoning. *Neuroimage*, 14(5):1136–1149, 2001. (cit. on pp. 61)

Sam J. Gilbert. Decoding the content of delayed intentions. *Journal of Neuroscience*, 31(8):2888–2894, 2011. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.5336-10.2011. URL https://www.jneurosci.org/content/31/8/2888. (cit. on pp. 61)

Ida Momennejad and John-Dylan Haynes. Human anterior prefrontal cortex encodes the 'what'and 'when'of future intentions. *Neuroimage*, 61(1):139–148, 2012. (cit. on pp. 61)

Ida Momennejad and John-Dylan Haynes. Encoding of prospective tasks in the human prefrontal cortex under varying task loads. *Journal of Neuroscience*, 33(44):17342–17349, 2013. (cit. on pp. 61)

James K Kroger, Fred W Sabb, Christina L Fales, Susan Y Bookheimer, Mark S Cohen, and Keith J Holyoak. Recruitment of anterior dorsolateral prefrontal

cortex in human reasoning: a parametric study of relational complexity. *Cerebral cortex*, 12(5):477–485, 2002. (cit. on pp. 61)

Silvia A Bunge, Carter Wendelken, David Badre, and Anthony D Wagner. Analogical reasoning and prefrontal cortex: evidence for separable retrieval and integration mechanisms. *Cerebral cortex*, 15(3):239–249, 2005. (cit. on pp. 61)

Adam E Green, Jonathan A Fugelsang, David JM Kraemer, Noah A Shamosh, and Kevin N Dunbar. Frontopolar cortex mediates abstract integration in analogy. *Brain research*, 1096(1):125–137, 2006. (cit. on pp. 61)

Adam Hampshire, Russell Thompson, John Duncan, and Adrian M Owen. Lateral prefrontal cortex subregions make dissociable contributions during fluid reasoning. *Cerebral cortex*, 21(1):1–10, 2011. (cit. on pp. 61)

Christine E Watson and Anjan Chatterjee. A bilateral frontoparietal network underlies visuospatial analogical reasoning. *Neuroimage*, 59(3):2831–2838, 2012. (cit. on pp. 61)

Aarit Ahuja, Theresa M. Desrochers, and David L. Sheinberg. A Role for Visual Areas in Physics Simulations. *Cognitive neuropsychology*, 38(7-8):425–439, 2021. ISSN 0264-3294. doi: 10.1080/02643294.2022.2034609. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9374848/. (cit. on pp. 61)

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018. (cit. on pp. 61)

Joseph R Manns and Howard Eichenbaum. Evolution of declarative memory. *Hippocampus*, 16(9):795–808, 2006. (cit. on pp. 61)

James C. R. Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy E. J. Behrens. The Tolman-Eichenbaum Machine: Unifying Space and Relational Memory through Generalization in the Hippocampal Formation. *Cell*, 183(5):1249–1263.e23, November 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.10.024. URL https://www.sciencedirect.com/science/article/pii/S009286742031388X. (cit. on pp. 61, 72)

Veronika Samborska, James L Butler, Mark E Walton, Timothy EJ Behrens, and Thomas Akam. Complementary task representations in hippocampus and prefrontal cortex for generalizing the structure of problems. *Nature Neuroscience*, 25(10):1314–1326, 2022. (cit. on pp. 61)

Stephanie Theves, David A Neville, Guillén Fernández, and Christian F Doeller. Learning and representation of hierarchical concepts in hippocampus and prefrontal cortex. *Journal of Neuroscience*, 41(36):7675–7686, 2021. (cit. on pp. 61)

Ricarda I Schubotz. Prediction of external events with our motor system: towards a new framework. *Trends in cognitive sciences*, 11(5):211–218, 2007. (cit. on pp. 62)

Jeffrey M Zacks. Neuroimaging studies of mental rotation: a meta-analysis and review. *Journal of cognitive neuroscience*, 20(1):1–19, 2008. (cit. on pp. 62)

Jason Fischer, John G Mikhael, Joshua B Tenenbaum, and Nancy Kanwisher. Functional neuroanatomy of intuitive physical inference. *Proceedings of the national academy of sciences*, 113(34):E5072–E5081, 2016. (cit. on pp. 62)

Nathaniel D Daw, Yael Niv, and Peter Dayan. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nature neuroscience*, 8(12):1704–1711, 2005. (cit. on pp. 62)

Anthony Dickinson and Bernard Balleine. The role of learning in the operation of motivational systems. *Stevens' handbook of experimental psychology*, 3: 497–533, 2002. (cit. on pp. 62)

Ray J Dolan and Peter Dayan. Goals and habits in the brain. *Neuron*, 80(2): 312–325, 2013. (cit. on pp. 62)

Geoffrey Schoenbaum, Matthew R Roesch, Thomas A Stalnaker, and Yuji K Takahashi. A new perspective on the role of the orbitofrontal cortex in adaptive behaviour. *Nature Reviews Neuroscience*, 10(12):885–892, 2009. (cit. on pp. 62)

David Badre, Andrew S Kayser, and Mark D'Esposito. Frontal cortex and the discovery of abstract action rules. *Neuron*, 66(2):315–326, 2010. (cit. on pp. 62)

David Badre and Michael J Frank. Mechanisms of hierarchical reinforcement learning in cortico–striatal circuits 2: Evidence from fmri. *Cerebral cortex*, 22(3):527–536, 2012. (cit. on pp. 62)

Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278, 2015. (cit. on pp. 62)

Jan Balaguer, Hugo Spiers, Demis Hassabis, and Christopher Summerfield. Neural Mechanisms of Hierarchical Planning in a Virtual Subway Network. *Neuron*, 90(4):893–903, May 2016. ISSN 0896-6273. doi: 10.1016/j.neuron.2016.03.037. URL https://www.cell.com/neuron/abstract/S0896-6273(16)30057-5. Publisher: Elsevier. (cit. on pp. 62)

Gasser Auda and Mohamed Kamel. Modular neural networks: a survey. *International Journal of Neural Systems*, 09(02):129–151, April 1999. ISSN 0129-0657. doi: 10.1142/S0129065799000125. URL https://www.worldscientific.com/doi/abs/10.1142/S0129065799000125. Publisher: World Scientific Publishing Co. (cit. on pp. 65)

Louis Kirsch, Julius Kunze, and David Barber. Modular Networks: Learning to Decompose Neural Computation, November 2018. URL http://arxiv.org/abs/1811.05249. arXiv:1811.05249 [cs, stat]. (cit. on pp. 65, 67, 73)

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing Networks: Adaptive Selection of Non-linear Functions for Multi-Task Learning, December 2017. URL http://arxiv.org/abs/1711.01239. arXiv:1711.01239 [cs]. (cit. on pp. 65, 67, 73)

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing Networks and the Challenges of Modular and Compositional Computation, April 2019. URL http://arxiv.org/abs/1904.12774. arXiv:1904.12774 [cs, stat]. (cit. on pp. 65, 74)

Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta Module Network for Compositional Visual Reasoning, November 2020. URL http://arxiv.org/abs/1910.03230. arXiv:1910.03230 [cs]. (cit. on pp. 65, 74)

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-term Memory. *Neural computation*, 9:1735–80, December 1997. doi: 10.1162/neco.1997.9.8.1735. (cit. on pp. 69)

Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, 21(6):860–868, June 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0147-8. URL https://www.nature.com/articles/s41593-018-0147-8. Number: 6 Publisher: Nature Publishing Group. (cit. on pp. 69)

Randall C. O'Reilly and Michael J. Frank. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, 18(2):283–328, February 2006b. ISSN 0899-7667. doi: 10.1162/089976606775093909. (cit. on pp. 69)

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines, December 2014. URL http://arxiv.org/abs/1410.5401. arXiv:1410.5401 [cs]. (cit. on pp. 69, 76, 84)

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, October 2016. ISSN 1476-4687. doi: 10.1038/nature20101. URL https://www.nature.com/articles/nature20101. Number: 7626 Publisher: Nature Publishing Group. (cit. on pp. 69, 70, 79, 84, 85)

Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z. Leibo, Adam Santoro, Mevlana Gemici, Malcolm Reynolds, Tim Harley, Josh Abramson, Shakir Mohamed, Danilo Rezende, David Saxton, Adam Cain, Chloe Hillier, David Silver, Koray Kavukcuoglu, Matt Botvinick, Demis Hassabis, and Timothy Lillicrap. Unsupervised Predictive Memory in a Goal-Directed Agent, March 2018. URL http://arxiv.org/abs/1803.10760. arXiv:1803.10760 [cs, stat]. (cit. on pp. 69)

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning, November 2016. URL http://arxiv.org/abs/1611.02779. arXiv:1611.02779 [cs, stat]. (cit. on pp. 69, 71)

Ignacio Cases, Clemens Rosenbaum, Matthew Riemer, Atticus Geiger, Tim Klinger, Alex Tamkin, Olivia Li, Sandhini Agarwal, Joshua D. Greene, Dan Jurafsky, Christopher Potts, and Lauri Karttunen. Recursive Routing Networks: Learning to Compose Modules for Language Understanding. In *Proceedings of the 2019 Conference of the North*, pages 3631–3648, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1365. URL http://aclweb.org/anthology/N19-1365. (cit. on pp. 69)

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to Ponder, September 2021. URL http://arxiv.org/abs/2107.05407. arXiv:2107.05407 [cs]. (cit. on pp. 70)

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, January 1999. ISSN 1546-1726. doi: 10.1038/4580. URL https://www.nature.com/articles/nn0199_79. Number: 1 Publisher: Nature Publishing Group. (cit. on pp. 70)

Louis Kirsch and Jürgen Schmidhuber. Meta Learning Backpropagation And Improving It, March 2022. URL http://arxiv.org/abs/2012.14905. arXiv:2012.14905 [cs, stat]. (cit. on pp. 71)

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to Learn Using Gradient Descent. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, Lecture Notes in Computer Science, pages 87–94, Berlin, Heidelberg, 2001. Springer. ISBN 978-3-540-44668-2. doi: 10.1007/3-540-44668-0_13. (cit. on pp. 71)

Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn, January 2017. URL http://arxiv.org/abs/1611.05763. arXiv:1611.05763 [cs, stat]. (cit. on pp. 71)

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables, March 2017. URL http://arxiv.org/abs/1611.00712. arXiv:1611.00712 [cs, stat]. (cit. on pp. 73)

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax, August 2017. URL http://arxiv.org/abs/1611.01144. arXiv:1611.01144 [cs, stat]. (cit. on pp. 73)

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, Montreal Quebec Canada, June 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553380. URL https://dl.acm.org/doi/10.1145/1553374.1553380. (cit. on pp. 75)

Guangyu Robert Yang, Madhura R. Joglekar, H. Francis Song, William T. Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, February 2019. ISSN 1546-1726. doi: 10.1038/s41593-018-0310-2. URL https://www.nature.com/articles/s41593-018-0310-2. Number: 2 Publisher: Nature Publishing Group. (cit. on pp. 76, 83, 85, 140)

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to Solve Arithmetic Word Problems with Verb Categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1058. URL https://aclanthology.org/D14-1058. (cit. on pp. 76)

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. Lila: A Unified Benchmark for Mathematical Reasoning, March 2023. URL http://arxiv.org/abs/2210.17517. arXiv:2210.17517 [cs]. (cit. on pp. 76)

Joshua Albrecht, Abraham J. Fetterman, Bryden Fogelman, Ellie Kitanidis, Bartosz Wróblewski, Nicole Seo, Michael Rosenthal, Maksis Knutins, Zachary Polizzi, James B. Simon, and Kanjun Qiu. Avalon: A Benchmark for RL

Generalization Using Procedurally Generated Worlds, October 2022. URL http://arxiv.org/abs/2210.13417. arXiv:2210.13417 [cs]. (cit. on pp. 76)

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic Curriculum Learning For Deep RL: A Short Survey, May 2020. URL http://arxiv.org/abs/2003.04664. arXiv:2003.04664 [cs, stat]. (cit. on pp. 77)

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014. URL http://arxiv.org/abs/1406.1078. arXiv:1406.1078 [cs, stat]. (cit. on pp. 79, 139)

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Publisher: IEEE. (cit. on pp. 82, 140)

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5), 2014. (cit. on pp. 82, 140)

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks, December 2015. URL http://arxiv.org/abs/1502.05698. arXiv:1502.05698 [cs, stat]. (cit. on pp. 84)

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the World State with Recurrent Entity Networks, May 2017. URL http://arxiv.org/abs/1612.03969. arXiv:1612.03969 [cs]. (cit. on pp. 84)

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers, March 2019. URL http://arxiv.org/abs/1807.03819. arXiv:1807.03819 [cs, stat]. (cit. on pp. 84, 85, 88)

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You

Need, August 2017. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs]. (cit. on pp. 85)

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum Learning: A Survey, April 2022. URL http://arxiv.org/abs/2101.10382. arXiv:2101.10382 [cs]. (cit. on pp. 94)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (cit. on pp. 123)