Exploring the role of (self-)attention in cognitive and computer vision architecture

Mohit Vaishnav

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 *13/04/2023* **par :**

Mohit VAISHNAV

Exploring the role of (self-)attention in cognitive and computer vision architecture

---

JURY

TIMOTHÈE MASQUELIER	CerCo, France	Président du Jury
JONATHAN D. COHEN	Princeton University, USA	Membre du Jury
HUGUES TALBOT	CentraleSupèlec, France	Membre du Jury
JESSICA B. HAMRICK	DeepMind, UK	Membre du Jury
THOMAS SERRE	Brown University, USA	Thesis Director
NICHOLAS ASHER	ANITI, France	Thesis Co-Director

École doctorale et spécialité :

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

Unité de Recherche :

Département Sciences et technologies de l'information et de la communication

Directeur(s) de Thèse :

Thomas SERRE et *Nicholas ASHER*

Rapporteurs :

Jonathan D. Cohen (Princeton University, USA) et *Hugues Talbot (CentraleSupèlec, France)*

RÉSUMÉ

Un mécanisme fondamental de la cognition, nécessaire à l'exécution de tâches de raisonnement complexes, est la capacité de traiter sélectivement les informations (attention) et de les conserver dans un état accessible (mémoire). Nous analysons systématiquement le rôle de ces deux composantes, en commençant par l'auto-attention basée sur le modèle d'attention le plus populaire: Transformer, et en étendant ensuite l'architecture à la mémoire. Transformer est aujourd'hui la dernière classe d'architecture neuronale et est au coeur des demonstrations les plus fascinante du Deep Learning, il a apporté un changement de paradigme dans le domaine de l'intelligence artificielle. Il a remplacé les réseaux de récurrence et de convolution par l'auto-attention comme choix architectural de facto pour la plupart des applications de l'IA.

Nous étudions d'abord les mécanismes de calcul impliqués dans un test de raisonnement visuel synthétique (SVRT), en analysant la capacité d'une architecture de vision par ordinateur populaire (ResNet) de différentes profondeurs et entraînée sur des ensembles de données de différentes tailles. Cela a conduit à une nouvelle taxonomie plus fine pour les vingt-trois tâches de SVRT, cohérente avec les classes de tâches de raisonnement - identiques-différentes (SD) et de relations spatiales (SR) - largement acceptées dans la littérature. Ensuite, nous étudions le rôle de l'auto-attention incorporée à ResNet50 dans la résolution du défi SVRT. Inspirés par les deux types de systèmes d'attention visuelle, nous avons modélisé l'auto-attention pour qu'elle soit utilisée comme une attention basée sur les caractéristiques et sur une attention spatiale pour enrichir les cartes de caractéristiques d'un réseau feedforward. Nous avons évalué la capacité de ces réseaux d'attention à résoudre le défi SVRT et avons constaté que les architectures résultantes étaient beaucoup plus efficaces pour résoudre la plus difficile de ces tâches de raisonnement visuel. La nouvelle taxonomie obtenue précédemment s'explique aussi partiellement par l'amélioration relative des deux réseaux d'attention et conduit à des prédictions testables concernant les besoins attentionnels des tâches SVRT.

Enfin, nous développons une nouvelle architecture cognitive intégrant l'auto-attention et la mémoire. Nous proposons GAMR (**G**uided **A**ttention **M**odel for visual **R**easoning), motivé par la théorie de la vision active. Le GAMR a des mécanismes de fonctionnement similaires à ceux du cerveau qui résout des tâches complexes de raisonnement visuel par des séquences de changements d'attention pour sélectionner et acheminer en mémoire les informations visuelles pertinentes pour la tâche. Ce changement d'attention est mis en œuvre à l'aide d'un module d'auto-attention guidé par une requête générée en interne. Nous démontrons

que *GAMR* est efficace, robuste et compositionnel par rapport à l'une ou l'autre des architectures basées sur le feedforward, l'attention ou la mémoire. De plus, GAMR est capable de généraliser à des tâches de raisonnement complètement nouvelles. Dans l'ensemble, notre travail analyse le rôle de l'auto-attention dans l'architecture cognitive et de vision par ordinateur par leur capacité á résoudre des tâches complexes de raisonnement visuel nécessitant de l'attention comme élément clé pour résoudre efficacement les tâches de raisonnement.

ABSTRACT

A fundamental mechanism of cognition needed to perform complex reasoning tasks is the ability to selectively process information (attention) and retain information in an accessible state (memory). We systematically analyze the role of both these components, starting with Transformer-based self-attention as a model of attention and later extending the architecture with memory. The Transformer is the latest and seemingly most powerful class of neural architecture, and it has brought a paradigm shift in the field of artificial intelligence. It has replaced recurrence and convolution networks with self-attention as the de-facto architectural choice for most AI applications.

We first study the computational mechanisms involved in a synthetic visual reasoning test (SVRT) challenge, analyzing the ability of popular computer vision architecture (ResNet) of different depths trained on different dataset sizes. It led to a novel, finer taxonomy for the twenty-three SVRT tasks consistent with the broadly accepted same-different (SD) and spatial-relation (SR) classes of reasoning tasks in literature. Next, we study the role of self-attention incorporated with ResNet50 in solving the SVRT challenge. Inspired by the two types of visual attention systems, we modeled self-attention to be used as feature-based and spatial attention to enrich the feature maps of a feedforward network. We evaluated the ability of these attention networks to solve the SVRT challenge and found the resulting architectures to be much more efficient at solving the hardest of these visual reasoning tasks. The novel taxonomy obtained earlier is also partially explained by the relative improvement of the two attention networks and leads to testable predictions regarding the attentional needs of SVRT tasks.

At last, we develop a novel cognitive architecture integrating self-attention and memory. We propose GAMR (**G**uided **A**ttention **M**odel for visual **R**easoning), motivated by the theory of active vision. GAMR has similar working mechanisms as that of the brain that solves complex visual reasoning tasks via sequences of attention shifts to select and route the task-relevant visual information into memory. This shift of attention is implemented with the help of a self-attention module guided by an internally generated query. We demonstrate that *GAMR* is sample-efficient, robust, and compositional compared to either of the feedforward, attention or memory-based architectures. In addition, GAMR is shown to be capable of zero-shot generalization on completely novel reasoning tasks. Overall, our work analyzes the role of self-attention in cognitive and computer vision architecture by their ability to solve complex visual reasoning tasks needing attention as a key component to efficiently solve reasoning tasks.

To the former president, missile man of India, nuclear scientist, writer, poet, and educator
Dr. A. P. J. Abdul Kalam

ACKNOWLEDGMENTS

Sailing through the past three years has been an unforgettable experience filled with countless challenges. I would like to use this opportunity to show how grateful I am to all the people who have helped me throughout this exciting journey toward fulfilling my Ph.D.

First and foremost, I thank my academic advisor, Thomas Serre (Brown University, USA) and Nicholas Asher (ANITI, France), for accepting me at ANITI. Words cannot express my gratitude to them for their invaluable guidance and patience and for providing me with the intellectual freedom to work. I am immensely thankful to Thomas Serre for assisting me through this conjunction of neuroscience and AI. All the conversations we had together helped me to develop scientific thinking and to comprehend the ability to filter out the most exciting approaches to be followed. I admire his unwavering support towards my unpolished ideas and guidance to give them shape. One-to-one review meetings with him and his critical judgment helped to push my boundaries and enlightened me with countless ideas to move forward.

I acknowledge the members of my thesis committee, Jonathan D. Cohen (Princeton Neuroscience Institute, Princeton University, USA), Hugues Talbot (CentraleSupèlec, France), Jessica Hamrick (Senior Research Scientist, DeepMind, UK), and Timothèe Masquelier (Senior Research Scientist, CerCo, France)

Additionally, this endeavor would not have been possible without the Agence Nationale de la Recherche (ANR) support for generously financing my research. I sincerely thank Corinne Joffre, Secrètaire gènèrale, ANITI, and her colleagues for supporting my multiple mobilizations between the USA and France.

This journey would have been unfinished without the help of the computing staff at High-Performance Cluster (HPC) *Oscar*, Brown University, USA and *CALMIP*, Universitè Fèdèrale de Toulouse Midi-Pyrènèes (UFTMiP), France. They provided their expertise to help me handle computationally intensive jobs.

A special thanks to Rufin VanRullen, Research Director at *CerCo*, who mentored me with his highly reliable and practical scientific knowledge and offered an office space with his team. During my initial days, I shared the office space with Andrea Alamia and Aimen Zerroug, who later became my friends and collaborators. Aimen is also my travel mate, with whom I visited Brown University and numerous other places, and we brainstormed several ideas together. I met Rufin's wonderful *NeuroAI* team members Milad Mozafari, Romain Bielawaski, Bhavin Choksi, Javier Cuadrado, Mathieu Chalvidal, Benjamin Devillers, Colin Decourt, Ismail Khalfaoui, Sabine Muzellec. We exchanged scientific insights

during our lab meetings (omitting *Pizza* in those meetings will be a crime). Although the duration of my Ph.D. was amid the COVID epidemic, I had the chance to attend in-person summer school *SANDAL* organized by CerCo. Here I got the opportunity to have interdisciplinary discussions with other teams at CerCo.

I want to thank the members of the *Serre Lab*, Aimen Zerroug, Thomas Fel, Mathieu Chalvidal, Lakshmi N. Govindarajan, Jacob Rose, Pachaya Sailamul, Ivan Rodriguez, Rex Liu, Lore Goetschalckx, Victor Boutin, Drew Linsley. They were remarkably welcoming during my stay. They created a friendly and collaborative atmosphere and shared their vast knowledge. Their valuable feedback and insightful analyses enabled me to refine my work. It might be unfair if I do not mention the writable walls of the Carney Institute at Brown University. Great ideas were penned on it, accompanied by several hours of discussions. During my stay, I was fortunate to meet Ekta Tiwari, Anusha Allawala, Krishn Bera, and Mukesh Makwana from the CLPS department.

I am appreciative of Mohamed Kaaniche, Deputy Scientific Director at ANITI. We first met to organize and manage a conference at ANITI and transitioned to the idea of having student representatives. I got elected for the position along with Bertille Somon, Henri Trenquier, and Dana Pizzaro. We made an incredible team, and it was an opportunity for me to carve out my leadership and time management skills while continuing research.

I also had the opportunity to collaborate with Remi Cadene (Senior Scientist, Tesla), who encouraged me to organize my thoughts before working on them; Drew Linsley (Asst. Professor Research, Brown University), who inspired me with his choice of words in scientific writing and positive attitude for any new idea. I greatly benefited from the conversation with Jonathan D. Cohen (Professor, Princeton University) and his group members, especially Taylor W. Webb (University of California, Los Angeles). Their extensive discussions on the GAMR architecture helped me to comprehend better. I thank Peter Wilf (Professor, Pennsylvania State University) for having me on board with him to execute practical aspects of my understanding of Paleobotany. Lastly, our ongoing collaboration with Experimental Neurosurgery and Neuroanatomy at KU Leuven introduced me to another aspiring scientist, Jesus G. Ramirez. This opportunity allowed me to understand neural visual reasoning mechanisms at the anatomical level.

My mind and heart owe Ashwani Sharma (Asst. Professor, IIT Ropar), K. R. Ramakrishnan (Emeritus Prof, IISc Bangalore) and Anil Kumar Tiwari (Assoc. Professor, IIT Jodhpur), Ranjan Gangopadhyay (Emeritus Prof, IIT Kharagpur). They fueled my scientific curiosity during various stages of my life and supported me in becoming a keen researcher.

I would be remiss in not mentioning my family, especially my parents Smt. Vijaylakshmi Vaishnav and Shri. Bharat Kumar Vaishnav (Commandant, CRPF), sister Dr. Divya Vaishnav (Asst. Professor, Chandigarh University), Brother-In-Law Mr. Sunil Sharma (Senior Scientist, ISRO), and younger brother Mr. Gaurav Vaishnav (Provincial Civil Service, Govt. of Bihar). Their belief in me and endless moral support have kept my spirits and motivation high during this process.

Apart from family members, friends play a vital role in one's life. I could not have undertaken this journey without the support of my friends now settling in different parts of the world. Special thanks to Dr. Dinesh K. Chobey, Pragya Das, John Thompson, Mohit Ahuja, Parita Verma, Himanshu Vaishnav, Malav Bateriwala, Pragnya Paramita ..., too many to be named all.

I want to end with a quote representing the state of my mind:

The more I learn, the more I realize how much I don't know.

– Albert Einstein

CONTENTS

Résumé i

Abstract iii

Acknowledgments v

Contents vi

List of Figures ix

List of Tables xvi

1 Introduction 2
- 1.1 Self-attention-based Transformer architecture . . . 7
- 1.2 Self attention in vision tasks . . . 11
- 1.3 Transformer-based vision architecture . . . 14
- 1.4 Original Contributions . . . 18

2 Computational Demands of Visual Reasoning 21
- 2.1 Introduction . . . 22
- 2.2 Systematic analysis of SVRT tasks' learnability . . . 24
- 2.3 An SVRT taxonomy . . . 25
- 2.4 Conclusion . . . 29

3 Role of self-attention in a computer vision architecture 33
- 3.1 Introduction . . . 34
- 3.2 Experiment 1: Self-attention with ResNet50 . . . 35
- 3.3 Experiment 2: Feature vs. rule learning . . . 43
- 3.4 Conclusion . . . 46

4 Role of self-attention in a cognitive architecture 49
- 4.1 Introduction . . . 50
- 4.2 Related Work . . . 51
- 4.3 Proposed approach . . . 53
- 4.4 Hyperparameters . . . 57
- 4.5 Method . . . 59
- 4.6 Benchmarking guided attention . . . 61

4.7 Benchmarking the system 63
4.8 Learning Compositionality 64
4.9 Zero-shot generalization 65
4.10 Ablation Study 66
4.11 Additional Experiment 67
4.12 Conclusion and limitations 69

5 Discussion and Future work **74**

6 Publications **80**

Appendix **82**

A Synthetic Visual Reasoning Task **82**

B Computational Demands of Visual Reasoning **86**

References **93**

List of Figures

1.1 A summary of attention in Cognitive science and machine learning (source) . . . 5
1.2 Transformer architecture proposed by Vaswani et al. [2017] (source) 8
1.3 Illustration of Multi-head attention mechanism in a Transformer network (source) . . . 9
1.4 Vision Transformer architecture [Dosovitskiy et al., 2021] . . . 15
1.5 Computational demands for training Transformers vs. CNNs. Compute needed to train a Transformer network has increased by 275 times in the last two years. (source) . . . 17

2.1 Two SVRT sample tasks from a set of twenty-three in total. For each task, the leftmost and rightmost two examples illustrate the two categories to be classified. Representative samples for the complete set of twenty-three tasks can be found in Figure A1 and A2. 23
2.2 Test accuracy for each of the twenty-three SVRT tasks as a function of the number of training samples for ResNets with depths 18, 50 and 152, resp. The color scheme reflects the identified taxonomy of SVRT tasks (see Figure 2.3 and text for details). . . . 26
2.3 Dendrogram derived from an N-dim hierarchical clustering analysis on the test accuracy of N=15 ResNets[18/50/152] trained to solve each task over a range of training set sizes. . . . 27

3.1 Location of the Transformer self-attention modules in our ResNet extensions. . . . 37
3.2 Test accuracies for a baseline ResNet50 vs. the same architecture endowed with the two forms of attention for each of the twenty-three SVRT tasks when varying the number of training examples. A different axis scale is used for SR_2 to improve visibility. These curves are constructed by joining task accuracy for five points representing dataset sizes. . . . 39
3.3 Test accuracies for 50-layer ResNets with spatial attention (orange), feature-based attention (tan), or no attention (green). Each bar depicts performance after training from scratch on 10k samples. . . . 40

3.4 The benefit of attention in solving the SVRT is greatest in data-limited training regimes. The x-axis depicts the number of samples for training, and the y-axis depicts a ratio of the average performance of models with attention to models without attention. When the ratio is greater than 1, it shows that attention helps vs. hurts when lower than 1. This gives us five ratios per task and attention process corresponding to each dataset size. We performed a linear fitting procedure for these points and calculated the corresponding slope. This slope characterizes the relative benefits of attention for that particular task as the number of training examples available increases. If the benefit of attention is most evident in lower training regimes, one would expect a relatively small slope. If the benefit of attention is most evident in higher training regimes, one would expect a large slope. 41
3.5 Principal component analysis of the twenty-three tasks using the 15-dimensional feature vectors derived from Experiment 1 representing the test accuracy obtained for each task for different dataset sizes and ResNets of varying depths (18, 50 & 152). The dotted red line represents 4 different bins in which these tasks can be clustered. 42
3.6 Test accuracies for a baseline ResNet50 trained from scratch ("No initialization") vs. the same architecture pre-trained on an auxiliary task in order to learn visual representations that are already adapted to the SVRT stimuli for different numbers of training examples. The format is the same as used in Figure 3.2. A different axis scale is used for SR_2 to improve visibility. These curves are constructed by joining task accuracy for five points representing dataset sizes. 45

4.1 Our proposed *GAMR* architecture is composed of three components: an *encoder* module (f_e) builds a representation (z_{img}) of an image, a *controller* guides a transformer-based multi-head attention module to dynamically shift attention, and selectively route task-relevant object representations (z_t) to be stored in a memory bank (M). The recurrent controller (f_s) generates a query vector ($query_t$) at each time step to guide the next shift of attention based on the current fixation. After a few shifts of attention, a reasoning module (r_θ) learns to identify the relationships between objects stored in memory. 55
4.2 Encoder module (f_e) used in *GAMR*. It consists of four convolutional blocks to process input image of 128×128 resolution 55
4.3 **Abstract variable:** t-SNE plot of the output vector (*out*) obtained from the controller (f_e) for all 23 SVRT tasks independently. Each cluster can be clearly identified from other clusters representing different relations learned. Tasks are represented as labels with the same colored box around them placed at the mean location of the cluster. 56
4.4 Ablation on Multi-Head Attention. We analyzed the average performance on SVRT tasks by changing the number of heads (*1, 2, 4, 8, 16*) and 1k images used for training. On average, this effect is distinguishable in SD tasks while SR tasks are already at their ceiling. 57
4.5 **Benchmarking Guided Attention**: We compared the average accuracy over two sub-clusters of SVRT obtained by *GAMR* with its variant when we replaced the guided-attention module with the self-attention (*GAMR-SA*) and when we completely gave away attention and made it a relational reasoning architecture (*GAMR w/o Atn (RN)*). 62
4.6 Bar plot analysis for the SVRT tasks grouped in same-different (SD) and spatially related (SR) tasks. We compared the accuracies of five baseline architectures with *GAMR*. ResNet-50 ($ResNet$) has 23M parameters, Relation Network (RN) has 5.1M parameters, ResNet-50 with attention (*Attn-ResNet*) has 24M parameters and *GAMR* & *ESBN* both have 6.6M parameters. We trained these with .5k, 1k, 5k and 10k samples. 63

4.7 **Compositionality test**: We train the model with tasks containing specific rules (e.g., Task *1* representing same/different discrimination & task *10* involving identification if the four shapes form a square or not). We show that with its ability to compose already learned rules, *GAMR* can quickly learn with 10 samples per class to adapt to a novel scenario (e.g., *15* where the rule is to identify if the four shapes forming a square are identical or not.) 65
4.8 **Ablation studies**: We pruned separate parts of the model, one at a time, like the weighting factor at each time step (w_{k_t}), controller output (out), guided-attention (w_k), relational vector (all_{obj}), feature channel gain factor (g) and temporal context normalization (tcn) and show the variation in performance on SD and SR tasks when trained with 1k samples. 67
4.9 Ablation studies for *GAMR*. (a) We pruned separate parts of the model, one at a time, like the weighting factor at each time step (w_{k_t}), feature channel gain factor (g), controller output (out), relational vector (all_{obj}) and temporal context normalization (tcn) and show the variation in performance on tasks 1, 5, 7 and 2 when trained with 1k samples. 68
4.10 **Time steps**: Shift of attention with each time step in a task-dependent manner. In the first row, the task is to answer if the two shapes are touching each other from the outside. At each time step, the network explores the area where the shapes are touching each other. In the second row, tasks represented required to answer if one of the smaller shapes is inside a larger shape. The controller module for this task shifts attention across different shapes at each time step. 69
4.11 **ART for *GAMR*:** (a) Same/different discrimination task. (b) Relational match-to-sample task (answer is 2). (c) Distribution-of-three task (answer is 1). (d) Identity rules task (ABA pattern, answer is 3). 70

4.12 Test accuracy on ART with different holdout sets when the images are $centered$ and compare the accuracy when shapes are $jittered$ in every image. We find that unlike other baselines experiencing a huge drop in performance when shapes are jittered, GAMR is stable. We plot the average accuracy over ten runs on the dataset. x axis corresponds to the four types of tasks, and y represents the average accuracy score. These tasks are as follows: (a) same-different (SD) discrimination task, (b) Relation match to sample task (RMTS); (c) Distribution of three tasks (Dist3); and (d) Identity rule task (ID). 71
4.13 **ART:** Comparing the average performance of *GAMR* with other baselines over 10 runs for different holdout values (m = 0, 50, 85, 95). These models are evaluated on four types of tasks, i.e., Same-Different (SD), Relation match to sample (RMTS), Distribution of 3 (Dist3) and Identity rules (ID). 72

A1 Sample images for Same Different (SD) tasks 84
A2 Sample images for Spatial Relation (SR) tasks 85

B1 Slope attained by linear fitting of points obtained after taking the ratio of each of the network with spatial attention module and the test accuracy of a ResNet50 for each task and training condition for Same Different (SD) tasks 87
B2 Slope attained by linear fitting of points obtained after taking the ratio of each of the network with spatial attention module and the test accuracy of a ResNet50 for each task and training condition for Spatial Relation (SR) tasks 88
B3 Slope attained by linear fitting of points obtained after taking the ratio of each of the network with feature-based attention module and the test accuracy of a ResNet50 for each task and training condition for Same Different (SD) tasks 89
B4 Slope attained by linear fitting of points obtained after taking the ratio of each of the network with feature-based attention module and the test accuracy of a ResNet50 for each task and training condition for Spatial Relation (SR) tasks 90

B5 Test accuracies for a baseline ResNet50 trained from scratch ("No initialization") vs. the same architecture pre-trained on Imagenet data for different number of training examples. Also note that a different axis scale is used for SR_2 to improve visibility. 91

LIST OF TABLES

1.1 Complexity comparison of different networks for a sequence of length n and dimensionality d [Vaswani et al., 2017] 11

3.1 Pearson coefficient (r) and corresponding p values obtained by correlating the slope vectors of the spatial attention and the feature-based attention modules with the two principal components of Figure 3.5. See text for details. 43

4.1 **ART**: Number of training and test samples used for four different types of tasks. 58

4.2 **ART**: For four different tasks number of epochs and learning rates (LR) used to train different architectures. 59

4.3 Test accuracy to show if the model learns the correct rules when we train it with a task and test on a different set of SVRT tasks with *GAMR*, Attention with ResNet50 (Attn-ResNet) and ResNet-50 (ResNet). 66

A.1 Each cell represents attempts participants took to solve seven consecutive correct categorizations. Here, row and column represents $task\ number$ and $participant\ number$. Entries containing "X" indicate that the participant failed to solve the problem, and those cells are not included in the marginal means. [Fleuret et al., 2011] 83

Chapter 1

CHAPTER 1

INTRODUCTION

1.1 Self-attention-based Transformer architecture 7
1.2 Self attention in vision tasks 11
1.3 Transformer-based vision architecture 14
1.4 Original Contributions 18

Everyone knows what attention is. It is the taking possession by the mind, in clear and vivid form, of one out of what seems several simultaneously possible objects or trains of thought.

– William James

Attention is a field widely discussed and studied in neuroscience, psychology, cognitive science and machine learning [Chun et al., 2011, Cho et al., 2015]. Attention is the process of selectively focusing on a discrete aspect of information while ignoring other perceivable information. A widely accepted feature of attention is that it facilitates efficient use of the available computational resources.

The cognitive science literature depicts several aspects of attention, such as it can be concentrated, it can focus on a particular modality, it can be divided, it can be selective, and it can have a finite capacity. However, selectivity is its most characteristic feature. Selectivity is necessary because of the limited availability of resources. Recently visual attention has gained tremendous attention in the field of artificial intelligence. Visual attention [Ahmad, 1991] is the ability to prioritize the information while neglecting the irrelevant information to overcome the data overloading in our visual system. Visual attention helps in answering *what* to look and *where* to look. It has been vastly studied in psychology and neuroscience [Posner and Petersen, 1990, Bundesen, 1990, Desimone et al., 1995, Corbetta and Shulman, 2002, Petersen and Posner, 2012]. These studies have acted as a source of inspiration for several artificial intelligence models [Khosla et al., 2007, Lindsay and Miller, 2018, Vaishnav et al., 2022a, Vaishnav and Serre, 2022].

There are three categories of selectivity in a visual attention system: by spatial location *(space-based)* [Posner, 1980, Posner et al., 1982], by object membership *(object-based)* [Duncan, 1984, Egly et al., 1994a, Vecera and Farah, 1994, Kramer et al., 1997] and by particular features of the input *(feature-based)* [Harms and Bundesen, 1983, Driver and Baylis, 1989, Kramer and Jacobson, 1991, Baylis and Driver, 1992, Duncan and Nimmo-Smith, 1996].

Visual Spatial Attention Every second, our eye makes small and rapid movements several times, known as saccades. These eye movements change the locus of attention. Visible shifts of attention, such as saccades, are known as *overt* visual attention. One more method used to emphasize a spatial location without any over-the-shift of the fovea location is *covert* attention. An example is the subject's fixation on a particular region throughout a task where the stimulus is likely to appear. This region is also referred to as the "*spotlight*" of attention. Certain visual patterns that involve edges, contrast, or motion automatically attract attention. These patterns are known as "*salient*" [Itti and Koch, 2001]. In the presence of task-specific information, these saccadic movements are controlled in a top-down fashion around the particular visual target instead of the salient regions. Eye movements are one of the possible ways to control visual attention.

Visual Feature Attention When the focus of attention is on features like color, shape or orientation instead of location, it is known as feature-based attention. It is an example of covert visual attention. Cueing the right features enhances the system's performance. It is used in tasks such as visual search combining covert feature-based attention with overt attention. Feature-based attention is global as opposed to spatial attention, i.e., when attention is focused on a particular feature, neurons representing that particular feature in the visual space are also modulated [Saenz et al., 2002]. It is related to object attention, i.e., instead of attending to an abstract feature, the attention is deployed at a specific object in a visual scene [Chen, 2012]. A single feedforward pass in the visual hierarchy can segregate the objects of a visual scene if there is a distinct salient difference between them as opposed to a complex scene where recurrent and serial processing might be required [Lamme and Roelfsema, 2000].

In addition to feature-based or spatial attention, another widely accepted classification is characterized by the type of data processing [Connor et al., 2004, Buschman and Miller, 2007]. There are two types of data processing, *bottom-up* and *top-down*. In a bottom-up attention process, external factors guide the attentional process because of their inherent properties, like their color or sudden motion in the scene. It is fast and primitive sensory driven. In top-down attention, there is an internal attentional guidance mechanism based on prior knowledge and current goals, like searching for food if one is hungry. It can ignore the salient stimuli and focus on the target object or event.

Attention is also involved while performing tasks requiring multiple sensory

signals. In the presence of multiple tasks or sensory signals, the central executive controller helps to route the focus of attention. The Central executive controller is responsible for coordinating activity with the cognitive system for directing attention, decision making and maintaining task goals. Context and history are deemed helpful to executing tasks optimally – making it highly related to the working memory. Attention is furthermore seen as the output of the central controller. The controller selects the targets of attention and passes them to the system responsible for its implementation. There is a three-way relationship between executive control, working memory and attention in such a way that the focus of attention is selected by the executive controller based on the contents of the working memory [Soto et al., 2008]. Although all the objects in the working memory can influence attention, the executive controller helps decide which one should affect the most [Olivers and Eimer, 2011]. These vast and extensive cognitive studies related to attention have inspired the field of AI and helped to boost its performance (Figure 1.1).

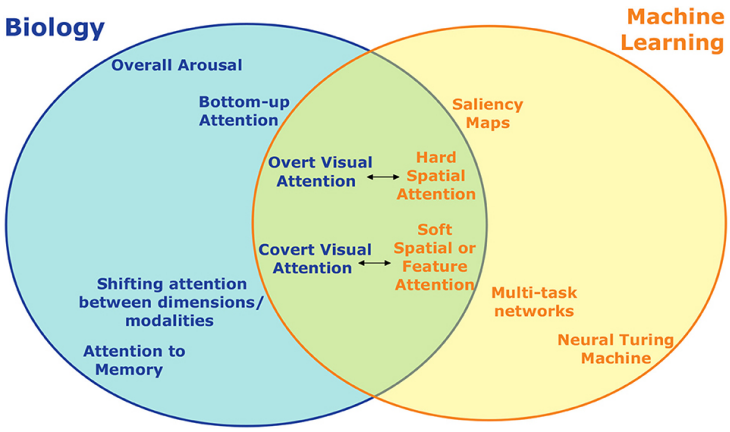


Figure 1.1: A summary of attention in Cognitive science and machine learning (source)

The first attempt to adapt the attention mechanism in a neural network was made in the 1980s when the improved version of Neocognitron [Fukushima, 1980] incorporated selective attention [Fukushima, 1987] to decompose the image into elementary features. Later, Fukushima and Imagawa [1993] modified the network to recognize and segment characters in cursive handwriting. Postma et al. [1997] proposed an attentional scanning model, SCAN, to attend to and identify object patterns without decomposing the scene into elementary features. As an alterna-

tive to these static neural approaches, Schmidhuber and Huber [1991] proposed a sequential model inspired by the sequential eye movements for object detection. In this model, a neural controller learns sequential generation of fovea trajectories to reach the target. Furthermore, data processing types inspired the development around the same time, thereby leading to a model extracting the region of interest using bottom-up and top-down processing Milanese et al. [1994].

By the early 2000s, the influence of attention on the evolution of neural networks increased. Miau and Itti [2001] proposed a model of primate vision integrating both, *what* and *where* pathways. The model has a fast visual attention-based frontend to select the most salient image areas and a slow backend to recognize objects in those selected areas. Another model based on the primate selective mechanism is presented in Salah et al. [2002] with the idea of selectively attending to relevant parts of the input image. In this model, a neural network analyzes the input image and generates posterior probabilities for the Markov models. Attention has also been used for object recognition [Walther et al., 2002] and scene analysis [Schill et al., 2001].

The year 2015 marks the new innings of attention-based architectures with the introduction of the attentional model for Neural Machine Translation (NMT) [Bahdanau et al., 2014a, Luong et al., 2015] and image captioning [Xu et al., 2015]. In NMT, the expectation is to learn continuous representations of variable-length sequences. *Recurrent neural networks* (RNNs) like LSTMs [Hochreiter and Schmidhuber, 1997a], GRUs [Cho et al., 2014a] and Quasi-RNNs [Bradbury et al., 2017] were some of the popular sequence models for representation learning at that time. While these RNNs' output depends on the previous elements in a sequence, traditional feedforward neural networks assume that inputs and outputs are independent of each other. Nonetheless, their limitation includes their inability to parallelize computations – making them slow during training and their fixed-size memory – bottleneck for long-range interactions [Vaswani et al., 2017].

Models used for NMT typically consist of encoder-decoder architecture [Cho et al., 2014b]. Typically, both encoder and decoder are RNNs, where the encoder takes an input sequence of fixed-length vector and represents it again to another fixed-length vector. A decoder then takes this encoded vector to generate the output sequence token by token. However, this method has two challenges; first, the encoder compresses the input sequence into a fixed vector length which may lead to the loss of information [Cho et al., 2014a]. Second, the model is incapable of aligning between input and output sequences which is essential for tasks such

as translation or summarization [Young et al., 2018]. While generating the output sequence, the decoder also lacked the mechanism to selectively focus on relevant input tokens. Later, Bahdanau et al. [2014b] proposed a sequence-to-sequence modeling task with the help of soft attention, emphasizing the parts of the sentence relevant to predicting the target word. Bahdanau et al. [2014b] extended the basic encoder-decoder by letting the model search a set of input words while generating target words. It allowed the model to focus on information needed to generate the subsequent target sequence.

In the following two years, the adoption of attentional mechanisms in neural networks diversified. Content-based soft attention mechanism [Goodfellow et al., 2016] is used in Neural Turing Machine (NTM) [Graves et al., 2014] with end-to-end training. Around the same time, Cheng et al. [2016] used a form of attention called intra-attention in the Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997b] architecture. [Hochreiter and Schmidhuber, 1997b] embedded a memory network inside the LSTM architecture to store the contextual representation of the input. This memory network has a set of key and value vectors in the hidden state to represent what is stored in the memory. These vectors are used to estimate the intra-attention with the previously stored tokens in the memory as opposed to the self-attention mechanism used by Vaswani et al. [2017] where interaction between the whole input sequence is estimated. One of the first uses of the self-attention mechanism in NLP is done by Parikh et al. [2016].

Since then, self-attention mechanisms have become an integral part of sequence modeling allowing the network to model dependencies between input and output sequences irrespective of their distances. A self-attention layer calculates a single-shot interaction between all pairs of words in a sequence.

1.1 Self-attention-based Transformer architecture

In 2017 Vaswani et al. [2017] proposed a novel architecture, *Transformer* for NLP. It is predominantly a self-attention network driving the waves of advances in AI. A Transformer architecture (Figure 1.2) includes a stack of encoder and decoder blocks. Each encoder block is identical and contains a self-attention layer and a feedforward layer. The encoder's input flows through the self-attention layer helping the encoder to look at other words while encoding the current word. Its output is then fed to the feedforward layer. The same feedforward network is

applied independently to each word. While a decoder consists of an encoder-decoder attention block in addition to the self-attention layer and a feedforward layer helping the decoder to focus on relevant parts of the input sequence.

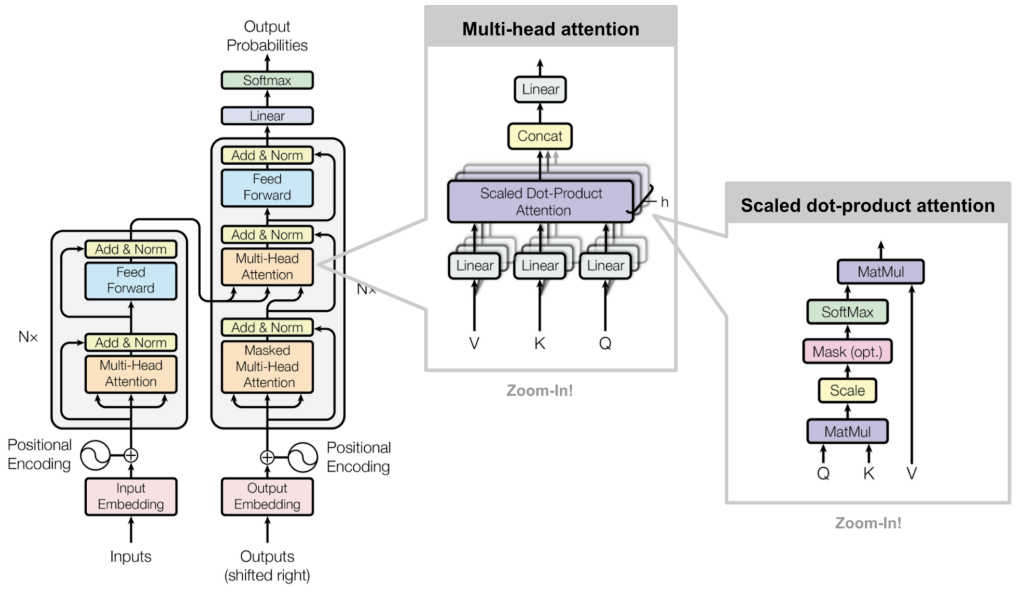


Figure 1.2: Transformer architecture proposed by Vaswani et al. [2017] (source)

In the NLP task, each word of the input sequence is first converted into an embedding vector. They are provided as input to the encoder block, passing through a self-attention layer and feedforward network. The obtained output vector is fed to the next encoder block. Using the self-attention layer, the Transformer models the relationship between the current word with other relevant words of a sequence.

In a self-attention layer, its input vector is transformed into a key (K), query (Q) and value (V) vectors of dimension $d_q = d_k = d_v = 512$ using a learnable matrix transformation. At first, the score (S) is calculated to determine the amount of focus to place on the other words in a sequence while encoding the current word. This score is calculated using the dot product between the query and key vectors ($S = Q \,.\, K^T$). It is normalized ($S = S/\sqrt{d_k}$) to stabilize the gradients, and later, using a $softmax$, converted into probabilities. The extent of the probability score shows the relevancy of the current word with other words in the sequence. This score is multiplied by the value vector (V) so that relevant words are given additional focus while irrelevant words are neglected in the subsequent layers.

$$Attention\ (Q, K, V)\ = softmax(\frac{Q\ .\ K^T}{\sqrt{d_k}})\ .\ V$$

The self-attention mechanism proposed by Vaswani et al. [2017] has an additional feature called *multi-head* attention (MHA). It helps to improve the performance in two ways: by augmenting the network's ability to focus on multiple positions and by giving distinct representational subspaces to each word. For example, if there are eight heads, eight sets of K, Q and V matrices exist, each representing a unique representational subspace. They are concatenated before passing through the feedforward network (Figure 1.3).

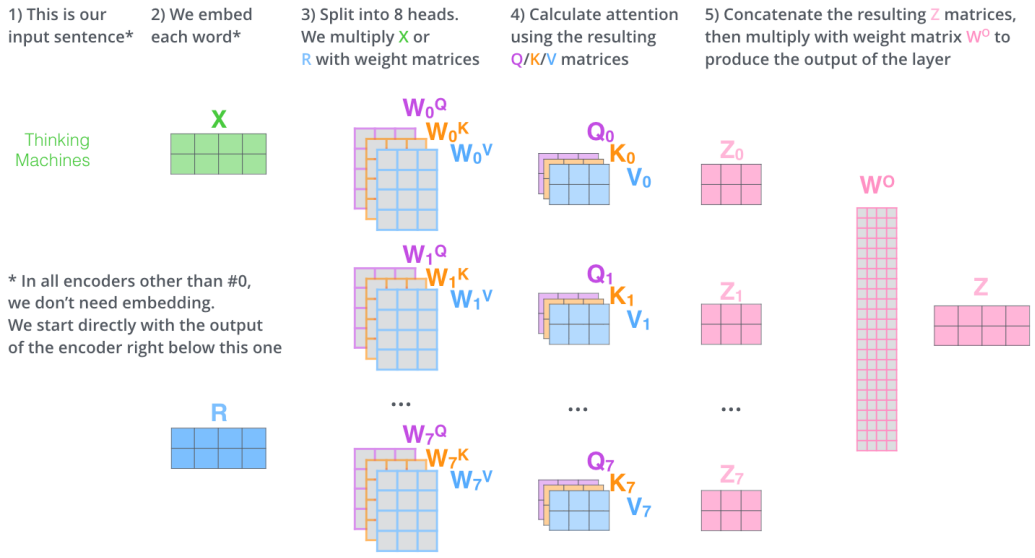


Figure 1.3: Illustration of Multi-head attention mechanism in a Transformer network (source)

The key characteristic of NLP tasks is the order of the words in a sequence. However, the operations we have discussed till now are permutation invariant. Positional embedding vectors are added to each input embedding vector to address this issue. These vectors help the model estimate the position of each word in a sequence in the projection space (i.e., $K/Q/V$).

Positional encoding in the Transformer is an active and vibrant research area. Vanilla Transformer uses absolute positional encoding; however, more recent work

[Devlin et al., 2018a, Dosovitskiy et al., 2021] prefers a learned [Gehring et al., 2017] or relative positional encoding [Shaw et al., 2018]. The absolute coordinate system does not encode translational equivariance, while relative geometry could. Ramachandran et al. [2019], Bello et al. [2019] studied different positional encoding techniques and established that relative positional encoding offers the best results while providing additional advantages like encoding for an unseen length of sequences (refer to Wu et al. [2021a] for a review). An overview of different positional encoding strategies used in NLP is discussed by Dufter et al. [2021].

The residual connection around the self-attention layer and a feedforward network is an essential module in an encoder block. It is followed by the layer normalization step [Baevski and Auli, 2018, Wang et al., 2019, Dosovitskiy et al., 2021]. A residual connection is added to each sub-layer in the encoder (and decoder) to strengthen the flow of information and achieve higher performance.

At the end of the encoding steps, decoding begins. The decoder uses the key (K) and value (V) vectors from the top-most encoder block for its encoder-decoder attention layer. It helps to focus on the appropriate locations of the input sequence. At each step, the decoder layer provides an element of the final output sequence. This output vector is again fed to the subsequent decoder layer in the next time step. This process continues till the end of the sequence. An independent set of positional encoding is applied on the decoder side.

The encoder-decoder attention module is similar to the multi-head self-attention mechanism described earlier. The only difference is that the key K and value V vectors are obtained from the top-most encoder block, and the query vector Q is derived from the previous self-attention layer of the decoder. Unlike the encoder, self-attention layers in the decoder are only allowed to access previously obtained output by masking the future words of a sequence. Masking future positions is done to prevent the decoder from cheating during the training phase – otherwise, it will already know what is coming next. The linear layer at the end of the decoder block is a fully connected neural network. It projects the vector obtained from the decoder layers into a logit vector. This logit vector represents the complete vocabulary of the language where translation has to be performed. A softmax converts this logit to the probabilities and represents the concerned word from the available vocabulary.

In terms of computational complexity, for a sequence of length n and dimensionality d, self-attention layers are faster than recursive or convolutional layers

when n is smaller than d, which is typically the case.

Layer Type	Complexity per Layer	Sequential operations	Maximum path length
Self-Attention	$\mathcal{O}(n^2.d)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Recurrent	$\mathcal{O}(n.d^2)$	$\mathcal{O}$(n)	$\mathcal{O}$(n)
Convolutional	$\mathcal{O}(k.n.d^2)$	$\mathcal{O}(1)$	$\mathcal{O}(log_k$(n))

Table 1.1: Complexity comparison of different networks for a sequence of length n and dimensionality d [Vaswani et al., 2017]

1.2 Self attention in vision tasks

Why self-attention for vision? In a self-attention mechanism, each word of a sequence is correlated with all the others. Thus containing information about the rest of the sequence – increasing the receptive field size equivalent to the length of a sequence. In some sense, images are no different from NLP sequences. Computer vision can take inspiration from the NLP domain to model long-range interactions between pixels with the added benefit of multi-head attention helping to parallelize these interactions. With the help of the multi-head attention method, different heads can focus on modeling different relations between pixels. For example, in a visual reasoning task where the objective is to count the number of pairs in an image, one head can focus on finding a pair while the other can focus on counting them. It helps the network to model self-similarity within an image. Images such as natural scenes and paintings display a great amount of self-similarity. Such non-local self similarity property was earlier explored for applications such as texture synthesis [Efros and Leung, 1999], object detection and segmentation [Wang et al., 2018], bilateral filtering [Tomasi and Manduchi, 1998] and image classification [Parmar et al., 2018a]. Hereafter, the main focus of this thesis will be computer vision.

Self-attention with CNN In a computer vision task, the resolution of the images could reach around 1000×1000 px. Applying a self-attention mechanism to all these pixels (10^6 in number) is computationally expensive because of the

quadratic complexity associated with the length of the sequence. Convolutional layers, on the other hand, do not have this bottleneck. However, they face trouble capturing long-range interactions because of their inability to scale up with the large receptive fields.

To address this problem, there are predominantly two approaches. The first is to reduce the self-attention operation cost to a linear scale. Aligned to this line of work, Ramachandran et al. [2019] proposed a pure stand-alone attention model for vision tasks by replacing the convolution operations with self-attention operations. Nonetheless, the self-attention operation used in this approach is local. Another similar linear attention variant Halo [Vaswani et al., 2021] uses block-wise local attention to improve speed and accuracy.

The second approach is to build hybrid CNN-Transformer architectures where the convolutions operations are used to encode the input image, and attention is applied to those encoded features. Srinivas et al. [2021] explored a hybrid combination of CNNs and multi-head self-attention (MHSA) models and showed that replacing the 3×3 kernal size convolutional layer in the bottleneck blocks of ResNet [He et al., 2016] with MHSA layers improved several CNN baselines. Interestingly, DETR [Carion et al., 2020] showed that concatenating the Transformer model at the end of the feature-extraction network is helpful for tasks like detection, localization, and segmentation.

There are four broad categories of research to incorporate self-attention mechanism with CNN, which are as follows:

Inserting few attention modules in between residual blocks: Along this line of work, Wang et al. [2018], Chen et al. [2018] proposed a non-local block similar to Ramachandran et al. [2019] and used them for video-based applications. In this network, features are gathered and propagated motivated by the squeeze and excite [Hu et al., 2018] network. As mentioned earlier, these methods only focus on the spatial dimension for calculating the non-local interaction, so Yue et al. [2018] added a correlation factor between the channels to improve the model effectiveness. Similarly, Shen et al. [2021] proposed a method to bring down the quadratic complexity of the self-attention mechanism to a linear scale. A unique way to incorporate self-attention with a feedforward network is demonstrated by Vaishnav et al. [2022a] where the intermediate features of the network are passed through the self-attention layer to find the global association. This attention is applied directly over the feature space in contrast to the previously used methods

of squeezing the feature vector dimensions to save computations.

Inserting attention modules at the end: Usually, such models have a front-end of convolutional block acting as a feature extraction module for self-attention block as back-end. These models are used for tasks like object detection and semantic segmentation. Huang et al. [2019] designed criss-cross attention that learns the complete image dependency recurrently for semantic segmentation tasks using dot-product attention. Moving away from this trend of using self-attention operations, Carion et al. [2020] proposed DETR architecture by placing a Transformer model as the back-end.

Replacing convolution layers by self-attention layers: Self-attention mechanism used in this line of research is primarily local in nature to decrease the computational demand associated with the increasing sequence length in an image which is directly proportional to total pixel count. Bello et al. [2019] made a unique attempt to augment the feature maps of convolutional layers with the self-attention modules. Feature maps obtained with the help of the self-attention module are concatenated with the feature maps of CNNs. They discovered that replacing all the feature maps of CNN with the feature maps of self-attention layers degrades the system's performance. Contrary to their finding, Ramachandran et al. [2019] came up with the architecture replacing all the convolution layers with a local self-attention layer and achieved better performance than a fully convolutional network on the image classification task.

In addition to these four categories of research, where the primary focus is on computer vision applications, cognitive studies also explore self-attention mechanisms. In one of the first studies by Whittington et al. [2022], neural representations of Hippocampal formation are related to the Transformer model. They did this correspondence with the help of the Tolman-Eichenbaum Machine Whittington et al. [2020], a model for hippocampal formation. This work showed that when recurrent positional encodings are used in Transformer, they replicate spatial representations of hippocampal formations like place cells and grid cells. To analyze from an attentional point of view, Vaishnav et al. [2022a] studied the role of a self-attention layer of the Transformer model in understanding visual reasoning tasks. This layer is used as a feature-based or spatial attention layer. A multi-head self-attention layer is significantly different from the other existing self-attention models where the span of attention in the dot-product mechanisms is local. They proposed a self-attention mechanism that could be applied globally over a feed-forward network's spatial or feature space. This method gives the network higher

representation power because of its ability to use multi-head attention. Recently, Vaishnav and Serre [2022] built a cognitive architecture inspired by the active vision literature relating to the shifting of the spotlight of attention. This attention routing is implemented with a controller module consisting of a self-attention module and an LSTM layer, which generates a query to guide the shifting. More studies in the NLP domain focus on relating language models to brain activations; however, a similar trend is yet to be seen in the computer vision domain.

These developments exploring the self-attention mechanisms propelled toward building a fully self-based attention architecture for computer vision applications. Evolutions in the NLP domain were vital in inspiring the fully self-attention-based architecture for vision tasks.

1.3 Transformer-based vision architecture

The first fully self-attention-based Transformer architecture is presented by Dosovitskiy et al. [2021]. It is known as Vision Transformer (ViT) (Figure 1.4). In this architecture, an input image is divided into a sequence of image patches called visual tokens and transforms those patches before passing them to the network. The core idea is to treat each pixel as a token and pass it to the Transformer network. However, with the increasing size of the number of pixels, attention cost scales quadratically, so patches of 16×16 pixels are used instead. Each patch is flattened and linearly projected to a vector of the desired dimension. As the network is agnostic to the positions of these patches w.r.t. the input image, position embeddings are added to learn the 2D structure. ViT learns this encoded structural information while training. A learnable class embedding token is also added at the beginning of the sequence. A class embedding token is inspired from Devlin et al. [2018b] that is learned along with other patches while training the network. This learnable token eventually helps to predict the classification label with the help of a multi-layer perceptron (MLP) head.

When the ViT is trained on a mid-sized dataset like ImageNet [Deng et al., 2009], outcomes are not impressive because of their lack of inductive biases such as translational equivariance and locality. ViT experiences difficulty learning image-specific inductive biases like a CNN, as the model never sees the complete 2D image during training but only a sequence of transformed patches. Such CNN-like biases are compensated by training the model with massive databases

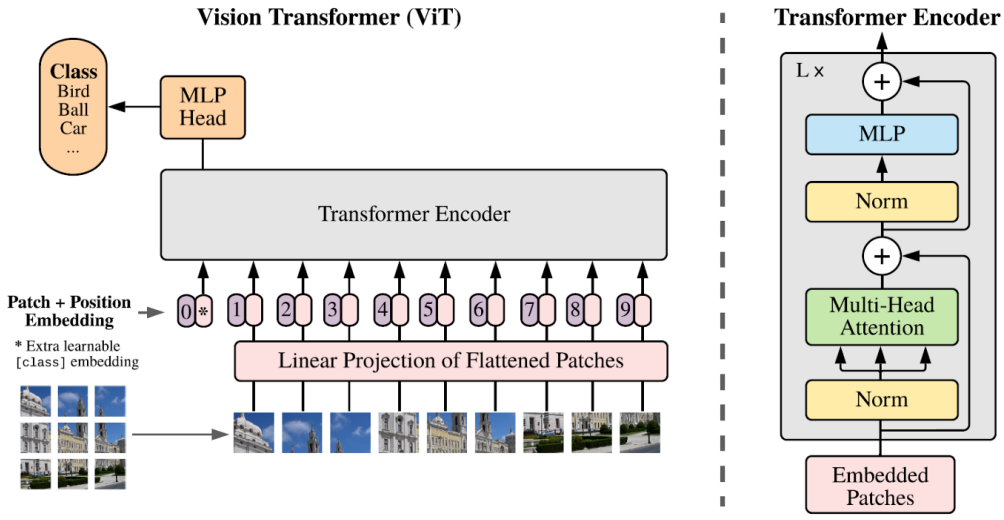


Figure 1.4: Vision Transformer architecture [Dosovitskiy et al., 2021]

like JFT-300M and fine-tuning it for downstream tasks. ViT learns the spatial relationship from scratch, which raises its demand for extra training data and longer training time. Touvron et al. [2021a] proposed DeIT, equalizing this pre-training-related bottleneck using techniques like the teacher-student distillation approach and robust augmentation methods. DeIT, when trained on ImageNet by incorporating these methods, surpasses the performance of the ViT model.

Vision transformers are a front-runner in capturing the long-range dependencies in an image, yet they fail to account for local features as CNNs do. A wide gap is perceived between ViT and CNN learnability. Wu et al. [2021b], Guo et al. [2022], Yuan et al. [2021], Graham et al. [2021], Dai et al. [2021], Peng et al. [2021] analysed the potential weaknesses in directly applying Transformer model from NLP domain and proposed a combination with convolutional network. Wu et al. [2021b] proposed a Convolutional vision Transformer (CvT) and presented a convolutional-based patch projection of image tokens along with a hierarchical design. Another alternative, LocalViT [Li et al., 2021] proposed depthwise convolution to capture local features. Meanwhile, LeViT Graham et al. [2021] enhanced the inference speed of ViT by designing multistage transformer architecture and downsampling the image using attention. In yet another network proposed by Zhou et al. [2021], it incorporated locality without convolutions with the help of enhanced local self-attention using Hadamard attention and ghost head.

Hadamard attention is more computational-friendly than dot-product attention, while ghost heads increase the channel capacity by combining attention maps.

A striking network, ConViT, proposed by d'Ascoli et al. [2021] took a step further to incorporate the convolutional biases into the Transformer architecture. d'Ascoli et al. [2021] initialized self-attention layers with soft convolutions with the help of Gated-Positional-Self-Attention (GPSA). This self-attention block is characterized by locality strength and head-specific center of attention. The locality factor determines how much the head should focus around its center of attention. For any given query patch, which head should give attention to which position is decided by the head-specific center of attention. With suitable parameters setting, ConViT can have ViT-like expressive power and could be trained in low-data regimes like CNNs. Recently, Vaishnav et al. [2022b] proposed *conviformer* to incorporate convolutional biases into any vision transformer with minimal architectural change. With the conviformer architecture, the network can also attend to higher-resolution images and provide compatibility with the base architecture used.

Challenges Transformer architecture confronts a two-front challenge. It requires enormous data for training to learn the right inductive bias and the computational cost associated with the sequence length. Figure 1.5 compares the computational requirement of different Transformers and CNNs models. An empirical study is done by Zhai et al. [2022] on the scalability of the ViT. They report that scaling up training samples and parameters of the model scale up the overall performance of the model; nevertheless, this plateaus quickly for smaller models as they cannot leverage additional data. It indicates that larger models have the scope to improve their representation learning abilities. Training a Transformer model requires massive data to compete with inductive biases like translation invariance similar to CNN. The self-attention mechanisms in a Transformer learn such image-specific concepts during longer training times, thereby significantly increasing the compute requirements. Strong data augmentation techniques nowadays compensate for the vast dataset requirement.

Transformer architecture furthermore lacks explicit mechanisms to attend to local neighborhoods. A commonly accepted solution to this issue is to restrict the attention mechanism to the local area Parmar et al. [2018b] or to incorporate structural priors on attention like Sparsity [Child et al., 2019]. It makes a dense attention matrix into a sparse matrix limiting the computations. Regardless, the

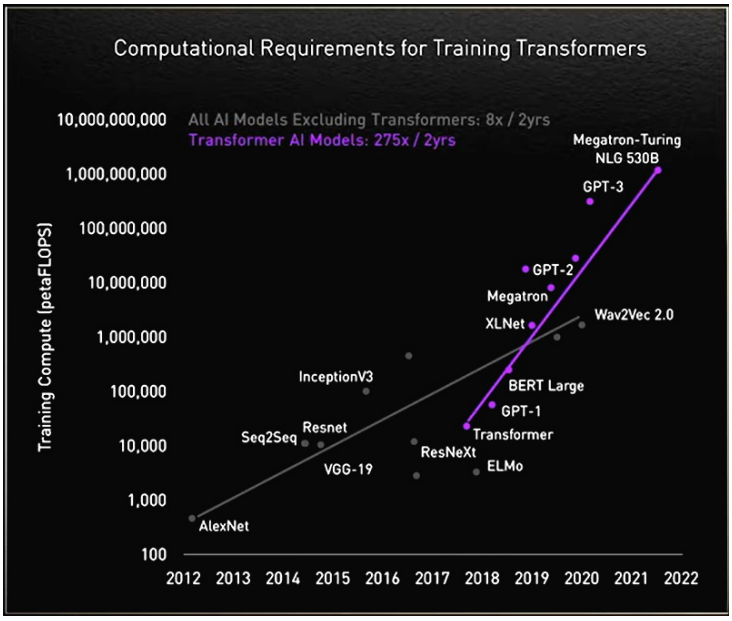


Figure 1.5: Computational demands for training Transformers vs. CNNs. Compute needed to train a Transformer network has increased by 275 times in the last two years. (source)

approach has some limitations. Sparse matrix multiplication operations are uncommon for hardware accelerators.

An additional computational bottleneck is calculating the dot product operation in the self-attention layer. Existing techniques to handle this situation are half-precision, gradient accumulation and gradient checkpointing. Tensor computations on modern hardware architectures are effectively done with 16-bit float tensors. Sometimes higher precision is required while calculating the loss, which doubles the required memory. This precision handling is carried out with the help of *apex* library[1]. On a fixed GPU/TPU machine, a large model may only fit a single-digit batch size, ultimately leading to unstable learning. A multivariate chain rule is used to incorporate the dynamics related to bigger batch sizes. It sums the gradients for a larger batch and computes the gradient descent at the end. For more bigger models, the trade-off is to separate the model into different chunks and compute the gradient in a forward/backward pass for each chunk.

[1] https://nvidia.github.io/apex/

Recently, Vaishnav et al. [2022b] proposed *conviformer* to address ViT's inability to process longer sequences which restricted ViT to smaller resolution images. In the conviformer, the input image is passed through a convolutional backbone, down-sampling the image to 224×224 (a commonly accepted input resolution). With the help of a convolutional frontend, the network makes sure to introduce the inductive biases of CNN into the network. The feature vectors obtained by the CNN modules are later passed to the base architecture of the vision transformer. This technique holds the compatibility of the network with the based model and provides a performance boost with insignificant additional computational cost.

Finally, training a huge Transformer model has negatively impacted the environment. Compute cost and the complexity associated with the Transformer are directly related to environmental factors such as CO_2 emission [Strubell et al., 2020] and high energy consumption [You et al., 2020]. There is also a cost associated with mining rare metals for manufacturing these hardware accelerators.

1.4 Original Contributions

Our contributions are as follows:

- We present a novel fine-grained taxonomy for the SVRT tasks by systematically analyzing the ability of feedforward neural networks.
- We first propose a self-attention-augmented feedforward network modeled as spatial or feature-based attention.
- Our attentional networks analysis on SVRT tasks provides a granular computational account of visual reasoning and yields testable neuroscience predictions regarding the differential need for feature-based versus spatial attention depending on the type of visual reasoning problem.
- Next, we present a novel transformer-based end-to-end trainable guided-attention module to learn to solve visual reasoning challenges in a data-efficient manner.
- We show that our multi-head transformer-based guided-attention module learns to shift attention to task-relevant locations and gate relevant visual

elements into a memory bank; such a multi-head transformer is also shown to perform significantly better than the popular self-attention mechanisms present in state-of-the-art transformer networks [Vaswani et al., 2017].

- We show that our architecture learns compositionally and is capable of learning efficiently by re-arranging previously-learned elementary operations stored within a reasoning module.
- Our architecture sets new benchmarks on SVRT [Fleuret et al., 2011] and ART [Webb et al., 2021], the two main visual reasoning challenges.

The work presented in Chapter 2 and Chapter 3 are taken from our following publication:

- **Mohit Vaishnav**, Remi Cadene, Andrea Alamia, Drew Linsley, Rufin VanRullen, Thomas Serre; "Understanding the Computational Demands Underlying Visual Reasoning." *Neural Computation* 2022; 34 (5): 1075–1099. doi: https://doi.org/10.1162/neco_a_01485

The work presented in Chapter 4 is taken from our following publication:

- **Mohit Vaishnav**, Thomas Serre. "GAMR: A Guided Attention model for (visual) Reasoning." *ArXiv* abs/2206.04928 (2022)

Chapter 2

CHAPTER 2

UNDERSTANDING THE COMPUTATIONAL DEMAND UNDERLYING VISUAL REASONING

2.1 Introduction 22
2.2 Systematic analysis of SVRT tasks' learnability 24
2.3 An SVRT taxonomy 25
2.4 Conclusion 29

2.1 Introduction

Humans can effortlessly reason about the visual world and provide rich and detailed descriptions of briefly presented real-life photographs [Fei-Fei et al., 2007], vastly outperforming the best current computer vision systems [Geman et al., 2015, Kreiman and Serre, 2020]. For the most part, studies of visual reasoning in humans have sought to characterize the neural computations underlying the judgment of individual relations between objects, such as their spatial relations (e.g., Logan [1994a]) or whether they are the same or different (up to a transformation, e.g., Shepard and Metzler [1971]). It has also been shown that different visual reasoning problems have different attentional and working memory demands [Logan, 1994b, Moore et al., 1994, Rosielle et al., 2002, Holcombe et al., 2011, Van Der Ham et al., 2012, Kroger et al., 2002, Golde et al., 2010, Clevenger and Hummel, 2014, Brady and Alvarez, 2015]. However, there is still little known about the neural computations that are engaged by different types of visual reasoning (see Ricci et al. [2021] for a recent review).

One benchmark that has been designed to probe abstract visual relational capabilities in humans and machines is the *Synthetic Visual Reasoning Test* (SVRT) [Fleuret et al., 2011]. The dataset consists of twenty-three hand-designed binary classification problems that test abstract relationships between objects posed on images of closed-contour shapes. Observers are never explicitly given the underlying rule for solving any given problem. Instead, they learn it while classifying positive and negative examples and receiving task feedback. Examples from two representative tasks are depicted in Figure 2.1: observers must learn to recognize whether two shapes are the same or different (Task *1*) or whether or not the smaller of the two shapes are near the boundary (Task *2*). Additional abstract relationships tested in the challenge include “inside", “in between”, “forming a square”, “aligned in a row" or “finding symmetry" (see Figures A1 and A2 for examples).

Most SVRT tasks are rapidly learned by human observers within twenty or fewer training examples [Fleuret et al., 2011] (see Table A.1; reproduced from the original study). On the other hand, modern deep neural network models require several orders of magnitude more training samples for some of the more challenging tasks [Ellis et al., 2015a, Kim et al., 2018, Messina et al., 2021a, Stabinger et al., 2021, 2016a, Puebla and Bowers, 2021] (see Ricci et al. [2021] for review; see also Funke et al. [2021a] for an alternative perspective).

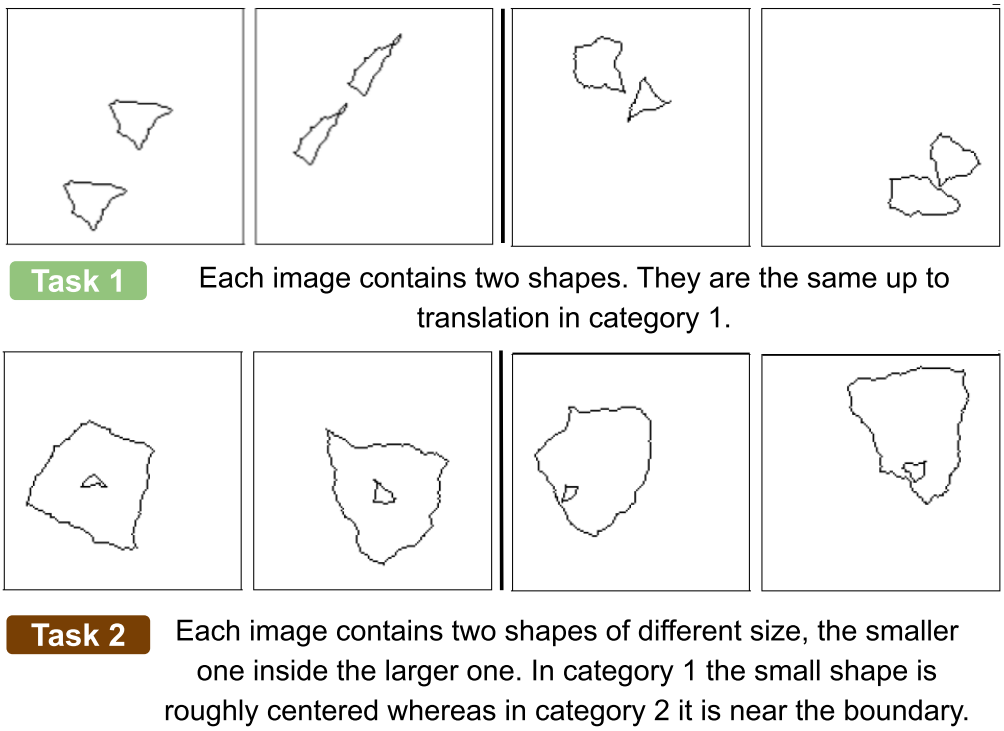


Figure 2.1: Two SVRT sample tasks from a set of twenty-three in total. For each task, the leftmost and rightmost two examples illustrate the two categories to be classified. Representative samples for the complete set of twenty-three tasks can be found in Figure A1 and A2.

It is now clear that some SVRT tasks are more difficult to learn than others. For instance, tasks that involve spatial-relation (SR) judgments can be learned much more easily by deep convolutional neural networks (CNNs) than tasks that involve same-different (SD) judgments [Stabinger et al., 2016a, Kim et al., 2018, Yihe et al., 2019a]. In contrast, a very recent study [Puebla and Bowers, 2021] demonstrated that even when CNNs learn to detect whether objects are the same or different, they fail to generalize over small changes in appearance, meaning that they have only partially learned this abstract rule. The implication of the relative difficulty of learning SR versus SD tasks is that CNNs appear to need additional computations to solve SD tasks beyond standard filtering, non-linear rectification, and pooling. Indeed, recent human electrophysiology work [Alamia et al., 2021a]

has shown that SD tasks recruit cortical mechanisms associated with attention and working memory processes to a greater extent than SR tasks. Others have argued that SD tasks are central to human intelligence [Firestone, 2020, Forbus and Lovett, 2021, Gentner et al., 2021a]. Beyond this basic dichotomy of SR and SD tasks, little is known about the neural computations necessary to learn to solve SVRT tasks as efficiently as human observers.

Here, we investigate the neural computations required for visual reasoning. In our experiment, we extend prior studies on the learnability of individual SVRT tasks by feedforward neural networks using a popular class of deep neural networks known as deep residual networks ("ResNets") [He et al., 2016]. We systematically analyze the ability of ResNets to learn all twenty-three SVRT tasks as a function of their expressiveness, parameterized by processing depth (number of layers), and their efficiency in learning a particular task. Through these experiments, we found that most of the performance variance in the space of SVRT tasks could be accounted for by two principal components, which reflected both the type of task (same-different vs. spatial-relation judgments) and the number of relations used to compose the underlying rules.

2.2 Systematic analysis of SVRT tasks' learnability

All experiments were carried out with the *Synthetic Visual Reasoning Test* (SVRT) dataset using code provided by the authors to generate images with dimension *128* $\times$ *128* pixels (see Fleuret et al. [2011] for details). All images were normalized and resized to 256$\times$256 pixels for training and testing models. No image augmentations were used during training. In our first experiment, we wanted to measure how easy or difficult each task is for ResNets to learn. We did this by recording the SVRT performance of multiple ResNets, each with different numbers of layers and trained with different numbers of examples. By varying model complexity and the number of samples provided to a model to learn any given task, we obtained complementary measures of the learnability of every SVRT task for ResNet architectures. In total, we trained 18-, 50-, and 152-layer ResNets separately on each of the SVRT's twenty-three tasks. Each of these models was trained with .5k, 1k, 5k, 10k, 15k, and 120k class-balanced samples. We also generated two unique sets of 40k positive and negative samples for each task: one was used as a validation set to select a stopping criterion for training the networks (if validation accuracy reaches 100%) and one was used as a test set to report model accuracy. In

addition, we used three independent random initializations of the training weights for each configuration of architecture/task and selected the best model using the validation set. Models were trained for *100* epochs using the $Adam$ optimizer [Kingma and Ba, 2014] with a training schedule (we used an initial learning rate of $1e$-3 and changing it to $1e$-4 from the 70^{th} epoch onward). As a control, because these tasks are quite different from each other, we also tested two additional initial learning rates (*1e-4, 1e-5*).

Consistent with prior work [Kim et al., 2018, Stabinger et al., 2016a, Yihe et al., 2019a], we found that some SVRT tasks are much easier to learn than others for ResNets (Figure 2.2). For instance, a ResNet50 needs only *500* examples to perform well on tasks *2, 3, 4, 8, 10, 11, 18* but the same network needs *120k* samples to perform well on task *21* (see Figures A1 and A2 for examples of these tasks). Similarly, with *500* training examples, task *2, 3, 4 & 11* can be learned well with only 18 layers while task *9, 12, 15 & 23* require as many as 152 layers. A key assumption of our work is that these differences in training set sizes and depth requirements between different SVRT tasks reflect different computational strategies that need to be discovered by the neural networks during training for different tasks. Our next goal is to characterize what these computational strategies are.

2.3 An SVRT taxonomy

To better understand the computational strategies needed to solve the SVRT, we analyzed ResNet performance on the tasks with a multi-variate clustering analysis. For each individual task, we created an N-dimensional vector by concatenating the test accuracy of all ResNet architectures ($N = 3$ depths $\times$ 5 training set sizes = 15), which served as a signature of each task's computational requirements. We then passed a matrix of these vectors to an agglomerative hierarchical clustering analysis (Figure 2.3) using the $Ward's$ method.

Our clustering analysis revealed a novel taxonomy for the SVRT. At the coarsest level, it recapitulated the dichotomy between *same-different* (SD; green branches) and *spatial-relation* (SR; brown branches) categorization tasks originally identified by Kim et al. [2018] using shallow CNNs. Interestingly, two of the tasks which were classified as SR by Kim et al. [2018] (tasks *6 & 17*) were assigned to the SD cluster in our analysis. We examined the descriptions of these two tasks as given in Fleuret et al. [2011] (see also Figures A1 and A2) and found that these

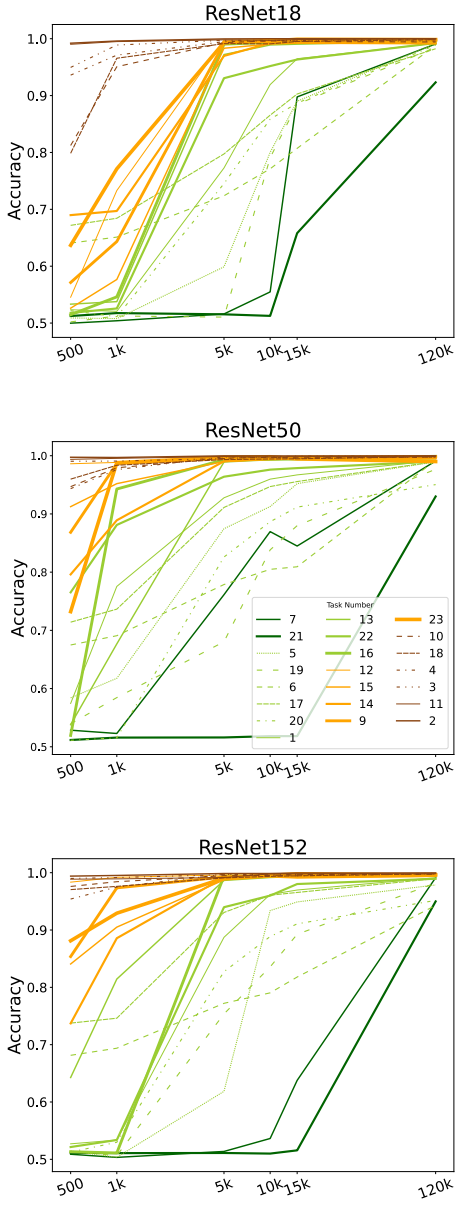


Figure 2.2: Test accuracy for each of the twenty-three SVRT tasks as a function of the number of training samples for ResNets with depths 18, 50 and 152, resp. The color scheme reflects the identified taxonomy of SVRT tasks (see Figure 2.3 and text for details).

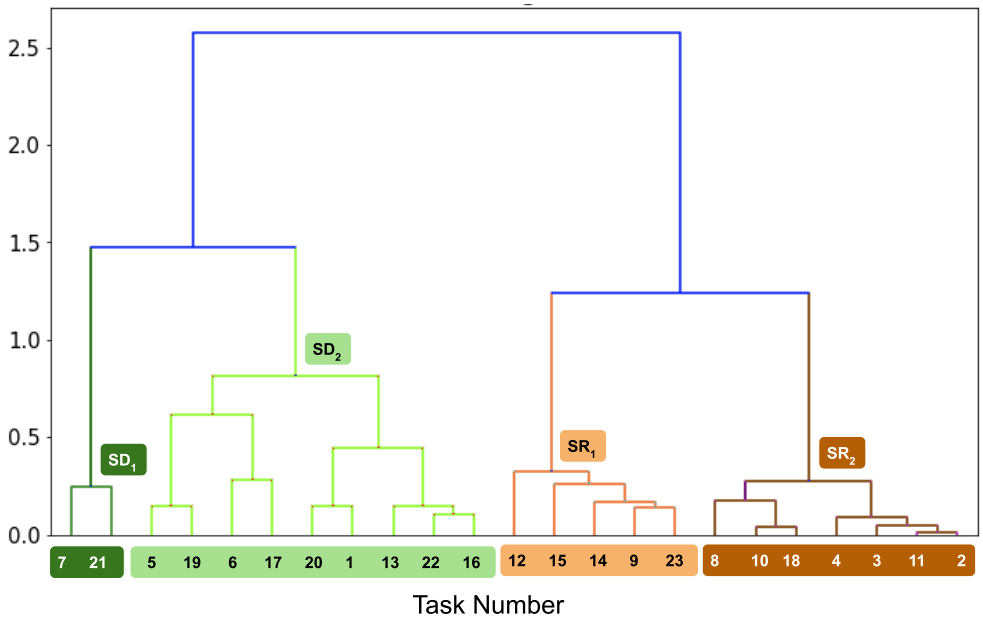


Figure 2.3: Dendrogram derived from an N-dim hierarchical clustering analysis on the test accuracy of N=15 ResNets[18/50/152] trained to solve each task over a range of training set sizes.

two tasks involve both SR and SD: they ask observers to tell whether shapes are the same or different and judge the distance between the shapes. Specifically, task *6* involves two pairs of identical shapes with one category having the same distance in-between two identical shapes vs. not in the other. Similarly, in task *17*, three of the four shapes are identical and their distance from the non-identical one is the same in one category vs. different in the other. Thus, our data-driven dichotomization of SR vs. SD refines the original proposal of Kim et al. [2018]. This could be due to our use of ResNets (as opposed to vanilla CNNs), deeper networks, and a greater variety of training set sizes (including much smaller training set sizes than those used by Kim et al. [2018]). The analysis by Fleuret et al. [2011] also revealed that several SD tasks (*6, 16, 17, 21*) are particularly challenging for human observers.

Our clustering analysis also revealed a finer organization than the main SR vs. SD dichotomy. The SR cluster could be further subdivided into two sub-clusters. The SR_2 (dark-brown-coloured) branch in Figure 2.3 captures tasks that involve

relatively simple and basic relation rules such as shapes making close contact (*3, 11*), or being close to one another (*2*), one shape being inside the other (*4*) or whether the shapes are arranged to form a symmetric pattern (*8, 10, 18*). In contrast, tasks that fall in the SR_1 (light-brown-colored) branch involve the composition of more than two rules such as comparing the size of multiple shapes to identify a subgroup before identifying the relationship between the members of the sub-groups. This includes tasks such as finding a *larger* object *in between* two smaller ones (*9*) or three shapes of which two are small and one large with two smaller (*identification of large and small object*) ones either inside or outside in one category vs. one *inside* and the other *outside* in the second (*23*), or *two small* shapes *equally close* to a bigger one (*12*), etc. These tasks also tend to be comparatively harder to learn, requiring ResNets with greater processing depth and more training samples. For instance, tasks *9, 12, 15, 23* were harder to learn than tasks *2, 4, 11* requiring more samples and/or more depth to solve well (Figure 2.2).

We found that task *15* gets assigned to this latter sub-cluster because the task requires finding four shapes in an image that are identical vs. not. One would expect this task to fall in the SD cluster but we speculate that the deep networks are actually able to leverage a shortcut [Geirhos et al., 2020] by classifying the overall pattern as symmetric/square (when the four shapes are identical) vs. trapezoid (when the four shapes are different; see Figure A2) – effectively turning an SD task into an SR task.

Our clustering analysis also reveals a further subdivision of the SD cluster. These tasks require recognizing shapes that are identical to at least one of the other shapes in the image. The first sub-cluster SD_2 (light green color branch) belongs to tasks that require identification of simple task rules, like answering whether or not two shapes are identical (even if it is along the perpendicular bisector) (tasks *1, 20*; see Figure A1), determining if all the shapes on an image are the same (*16, 22*), or detecting if two pairs of identical shapes can be translated to become identical to each other (*13*). Another set of tasks within this sub-cluster includes tasks that are defined by more complex rules that involve the composition of additional relational judgments. Sample tasks include identifying pairs/triplets of identical shapes and measuring the distance with the rest (*6, 17*), determining if an image consists of pairs of identical shapes (*5*), or detecting if one of the shapes is a scaled version of the other (*19*). Finally, the second sub-cluster SD_1 shown in dark-green color involves two tasks that require an understanding of shape transformations. One task asks observers to say if one of the shapes is the scaled, translated, or

rotated version of the other one (*21*). The other task test asks observers to judge whether or not an image contains two pairs of three identical shapes or three pairs of two identical shapes in an image (*7*).

To summarize this first set of experiments, we have systematically evaluated the ability of ResNets spanning multiple depths to solve each of the twenty-three SVRT tasks for different training set sizes. This allowed us to represent SVRT tasks with according to their learnability by ResNets of varying depth. By clustering these representations, we extracted a novel SVRT taxonomy that both recapitulated an already described SD-SR dichotomy [Kim et al., 2018], and also revealed a more granular task structure corresponding to the number of rules used to form each task. Tasks with more rules are harder for ResNets to learn. Our taxonomy also reveals an organization of tasks where easier SR_1 and SR_2 sub-clusters fall closer to each other than harder SD_1 and SD_2 sub-clusters.

2.4 Conclusion

The goal of the present study was to shed light on the computational mechanisms underlying visual reasoning using the Synthetic Visual Reasoning Test (SVRT) [Fleuret et al., 2011]. There are twenty-three binary classification problems in this challenge, which include a variety of same-different and spatial reasoning tasks.

In our experiment, we systematically evaluated the ability of a battery of $N = 15$ deep convolutional neural networks (ResNets) – varying in depths and trained using different training set sizes – to solve each of the SVRT problems. We found a range of accuracies across all twenty-three tasks. Shallower networks easily learned some tasks, and relatively small training sets and some tasks were hardly solved with much deeper networks and orders of magnitude more training examples.

Under the assumption that the computational complexity of individual tasks can be well characterized by the pattern of test accuracy across these $N = 15$ neural networks, we formed N-dimensional accuracy vectors for each task and ran a hierarchical clustering algorithm. The resulting analysis suggests a taxonomy of visual reasoning tasks: beyond two primary clusters corresponding to same-different (SD) vs. spatial relation (SR) judgments, we also identified a finer

organization with sub-clusters reflecting the nature and the number of relations used to compose the rules defining the task. Our results are consistent with previous work by Kim et al. [2018], who first identified a dichotomy between SD and SR tasks. Our results also extend prior work [Fleuret et al., 2011, Kim et al., 2018, Yihe et al., 2019a] in proposing a finer-level taxonomy of visual reasoning tasks. The accuracy of neural networks is reflected in the number of relationships used to define the basic rules, which is expected, but it deserves closer examination.

Kim et al. [2018] have previously suggested that SD tasks "strain" convolutional neural networks. That is, while it is possible to find a network architecture of sufficient depth (or the number of units) that can solve a version of the task up to a number of stimulus configurations (e.g., by forcing all stimuli to be contained within a $\Delta H \times \Delta W$ window), it is relatively easy to render the same task unlearnable by the same network past a certain number of stimulus configurations (e.g., by increasing the size of the window that contains all stimuli). It is as if these convolutional networks are capable of learning the task if the number of stimulus configurations remains below their memory capacity, and fails beyond that. It remains an open question whether non-convolution alternatives to the CNNs tested here such as the now popular transformer networks [Dosovitskiy et al., 2021, Touvron et al., 2021a, Tolstikhin et al., 2021] would learn to solve some of the harder SVRT tasks more efficiently. As an initial experiment, we attempted to train and test a Vision Transformer [1] (ViT) [Dosovitskiy et al., 2021] constrained to have a similar number of parameters (21M) to the ResNet-50 used here. We were not able to get these architectures to do well on most of the tasks that are difficult for ResNets, even with 100k samples (also shown in Messina et al. [2021b]). It is worth noting that even 100k samples remain a relatively small dataset size by modern-day standards since ViT was trained from scratch.

Multi-layer perceptrons and convolutional neural networks including ResNets and other architectures can be formally shown to be universal approximators under certain architectural constraints. That is, they can learn arbitrary mappings between images to class labels. Depending on the complexity of the mapping, one might need an increasing number of hidden units to allow for enough expressiveness of the network; but provided enough units / depth and a sufficient amount of training examples, deep CNNs can learn arbitrary visual reasoning tasks. While we cannot make any strong claim for the specific ResNet architectures used in this study (currently, the proof is limited to a single layer without max pooling or

[1] https://github.com/facebookresearch/dino

batch normalization [Lin and Jegelka, 2018]), we have indeed found empirically that all SVRT tasks could indeed be learned for networks of sufficient depth and provided a sufficient amount of training examples. However, deep CNNs typically lack many of the human cognitive functions, such as attention and working memory. Such functions are likely to provide a critical advantage for a learner to solve some of these tasks [Marcus, 2001]. CNNs might have to rely instead on function approximation which could lead to a less general "brute-force" solution. Given this, an open question is whether the clustering of SVRT tasks derived from our CNN-based analyses will indeed hold for human studies. At the same time, the prediction by Kim et al. [2018] using CNNs that SD tasks are harder than SR tasks and hence that they may demand additional computations (through feedback processes) such as attention and/or working memory was successfully validated experimentally by Alamia et al. [2021a] using EEG.

Additional evidence for the benefits of feedback mechanisms for visual reasoning was provided by Linsley et al. [2018a] who showed that contour tracing tasks that can be solved efficiently with a single layer of a recurrent CNN may require several order of magnitudes more processing stages in a non-recurrent-CNN to solve the same task. This ultimately translates into much greater sample efficiency for recurrent-CNNs on natural image segmentation tasks [Linsley et al., 2020]. The closely related task of "insideness" was also studied by Villalobos et al. [2021] who demonstrated the inability of CNNs to learn a general solution for this class of problems. Universal approximators with minimal inductive biases such as multi-layer perceptrons, CNNs and other feedforward or non-attentive architectures can learn to solve visual reasoning tasks, but they might need a very large number of training examples to properly fit. Hence, beyond simply measuring the accuracy of very deep nets in high data regimes (such as when millions of training examples are available), systematically assessing the performance of neural nets of varying depths and for different training regimes may provide critical information about the complexity of different visual reasoning tasks.

Chapter 3

CHAPTER 3

ROLE OF SELF-ATTENTION IN A COMPUTER VISION ARCHITECTURE

3.1 Introduction 34
3.2 Experiment 1: Self-attention with ResNet50 35
3.3 Experiment 2: Feature vs. rule learning 43
3.4 Conclusion 46

3.1 Introduction

Humans continue to outperform modern AI systems in their ability to flexibly parse and understand complex visual relations. Prior cognitive neuroscience work suggests that attention plays a key role in humans' visual reasoning ability. In the previous chapter, we discussed a benchmark used to evaluate the abilities of machines and compare them with humans. We did this by systematically assessing the ability of modern deep convolutional neural networks (CNNs) to learn to solve the synthetic visual reasoning test (SVRT) challenge, a collection of 23 visual reasoning problems. Our analysis revealed a novel taxonomy of visual reasoning tasks, which can be primarily explained by the type of relations (same-different (SD) versus spatial-relation (SR) judgments) and the number of relations used to compose the underlying rules.

Consistent with the speculated role of attention in solving the binding problem when reasoning about objects [Egly et al., 1994b, Roelfsema et al., 1998], prior work by Kim et al. [2018] has shown that combining CNNs with an oracle model of attention and feature binding (i.e., preprocessing images so that they are explicitly and readily organized into discrete object channels) renders SD tasks as easy to learn by CNNs as SR tasks. Here, we build on this work and introduce CNN extensions incorporating spatial or feature-based attention. In the first set of experiments, we show that these attention networks learn difficult SVRT tasks with fewer training examples than their non-attentive (CNN) counterparts but that the different forms of attention help on different tasks.

This experiment raises the question: how do attention mechanisms help with learning different visual reasoning problems? There are at least two possible computational benefits: attention could improve model performance by simply increasing its capacity, or attention could help models learn the abstract rules governing object relationships more efficiently. To adjudicate between these two possibilities, we measured the sample efficiency of ResNets pre-trained on SVRT images so that they only had to learn the abstract rules for each SVRT task. We found that attention ResNets and ResNets pre-trained on the SVRT were similarly sample-efficient in learning new SVRT tasks, indicating that attention helps discover abstract rules instead of merely increasing model capacity.

3.2 Experiment 1: Self-attention with ResNet50

We sought to identify computational mechanisms that could help ResNets learn the more challenging SVRT tasks revealed by our novel taxonomy. Attention has classically been implicated in visual reasoning in primates and humans [Egly et al., 1994b, Roelfsema et al., 1998]. Attentional processes can be broadly divided into ***spatial*** (e.g., attending to all features in a particular image location) vs. ***feature-based*** (e.g., attending to a particular shape or color at all spatial positions) [Desimone et al., 1995]. The importance of attention for perceiving and reasoning about challenging visual stimuli has also been realized by the computer vision community. There are now a number of attention modules proposed to extend CNN's – including spatial (e.g., Sharma et al. [2015], Chen et al. [2015], Yang et al. [2016], Xu and Saenko [2015], Ren and Zemel [2016]), feature-based (e.g., Stollenga et al. [2014], Chen et al. [2017], Hu et al. [2018]) and hybrid (e.g., Linsley et al. [2018b], Woo et al. [2018]) approaches. Here, we adapt the increasingly popular Transformer architecture [Vaswani et al., 2017] to implement both forms of attention. These networks, which were originally developed for natural language processing, are now pushing the state of the art in computer vision [Zhu et al., 2020, Carion et al., 2020, Dosovitskiy et al., 2021]. Recent work [Ding et al., 2021] has also shown the benefits of such architectures and especially attention mechanisms for solving higher-level reasoning problems.

Transformers are neural network modules usually consisting of at least one "self-attention" module followed by a feedforward layer. Here, we introduced different versions of the self-attention module into ResNets to better understand the computational demands of each SVRT task. Transformers' self-attention is applied to and derived from the module's input. By reconfiguring standard Transformer self-attention, we developed versions capable of allocating either spatial or feature-based attention over the input. Specifically, we created these different forms of attention by reshaping the convolutional feature map input to a Transformer. For spatial attention, we reshaped the $\mathcal{Z} \in \mathcal{R}^{H,W,C}$ feature maps to $\mathcal{Z} \in \mathcal{R}^{C,H*W}$ so that the Transformer's self-attention was allocated overall spatial locations. For feature-based attention, we reshaped the convolutional feature maps to $\mathcal{Z} \in \mathcal{R}^{H*W,C}$, enforcing attention to overall features instead of spatial locations.

Spatial Attention Module (SAM) Our first attention module takes a features map $X \in \mathcal{R}^{d_C \times d_H \times d_W}$ as input, where d_C, d_H, and d_W respectively refer to the number of channels, height and width of the map, and outputs a features map Y of the same dimensions. We flatten the spatial dimensions to obtain $X' \in \mathcal{R}^{d_C \times d_N}$, where $d_N = d_H \times d_W$, and we apply the original multi-head self-attention module from Vaswani et al. [2017] as follows.

We first apply independent linear mappings of the input X' to obtain three feature maps of dimensions $\mathcal{R}^{d \times d_N}$ for each attention head from a total of n_H heads. For the i^{th} head, these maps are known as the query Q_i, the key K_i and the value V_i, and are obtained such as:

$$Q_i = W_i^Q.X'$$
$$K_i = W_i^K.X'$$
$$V_i = W_i^V.X'$$

The mappings are parametrized by three matrices W_i^Q, W_i^K and W_i^V of dimensions $\mathcal{R}^{d \times d_C}$ for each head. The symbol . denotes a matrix multiplication.

Then, we apply the scaled dot-product attention [Vaswani et al., 2017] to obtain n_H attention heads of dimensions $\mathcal{R}^{d \times d_N}$ such as:

$$H_i = SoftMax(\frac{Q_i.K_i^T}{\sqrt{d}})V_i \tag{3.1}$$

After, we concatenate all attention heads along the first dimension and apply a linear mapping to obtain $Y' \in \mathcal{R}^{d_C \times d_N}$ such as:

$$Z = W^O.Concat(H_1, ..., H_{n_H}) \tag{3.2}$$

The mapping is parametrized by the matrix $W^O \in \mathcal{R}^{d_C \times d}$.

As commonly done, we have a residual connection before applying a layer normalization [Ba et al., 2016] such as:

$$Y' = LayerNorm(Z + X') \tag{3.3}$$

Finally, we unflatten Y' to obtain $Y \in \mathcal{R}^{d_C \times d_H \times d_W}$.

We obtain the best results with a representation space of 512 dimensions ($d = 512$) and four attention heads ($n_H = 4$).

Features-based Attention Module (FBAM) Our second attention module is simply obtained by transposing the channel dimension with the spatial dimensions before applying the same transformations. In other words, we transpose the input X' into $\mathcal{R}^{d_N \times d_C}$ and transpose the output Y' back into $\mathcal{R}^{d_C \times d_N}$. While SAM models attention over the $d_H * d_W$ regions that compose the input features map, FBAM models attention over the d_C features channels.

We obtain the best results with a representation space of 196 dimensions ($d = 196$) and one attention head ($n_H = 1$).

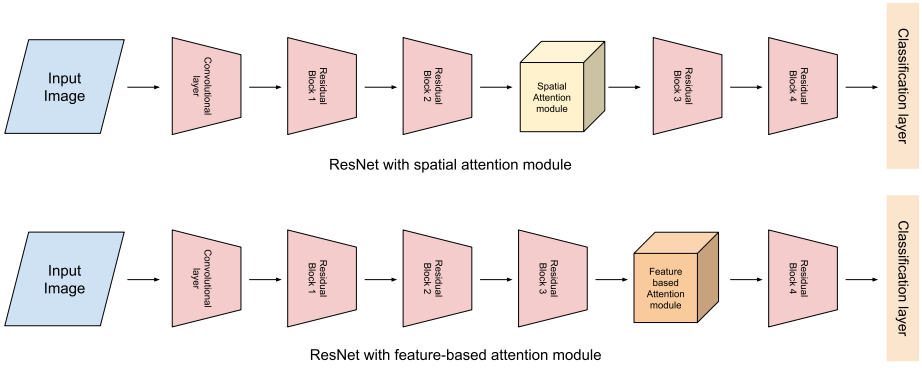


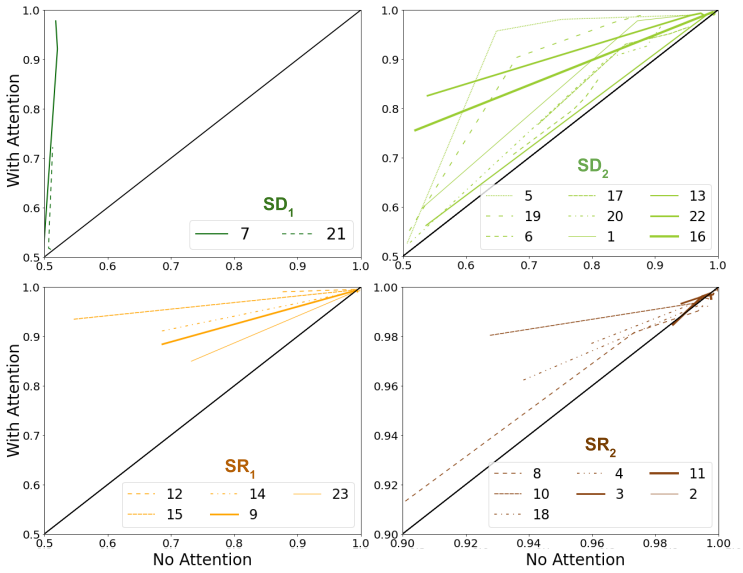
Figure 3.1: Location of the Transformer self-attention modules in our ResNet extensions.

We added one spatial or feature-based attention after one of the four residual blocks in a ResNet-50. We placed either form of attention module to a ResNet-50 by choosing the location where the addition of attention yielded the best validation accuracy across the SVRT tasks. Through this procedure, we inserted a spatial attention module after the second residual block and a feature-based attention module after the third residual block (Figure 3.1).

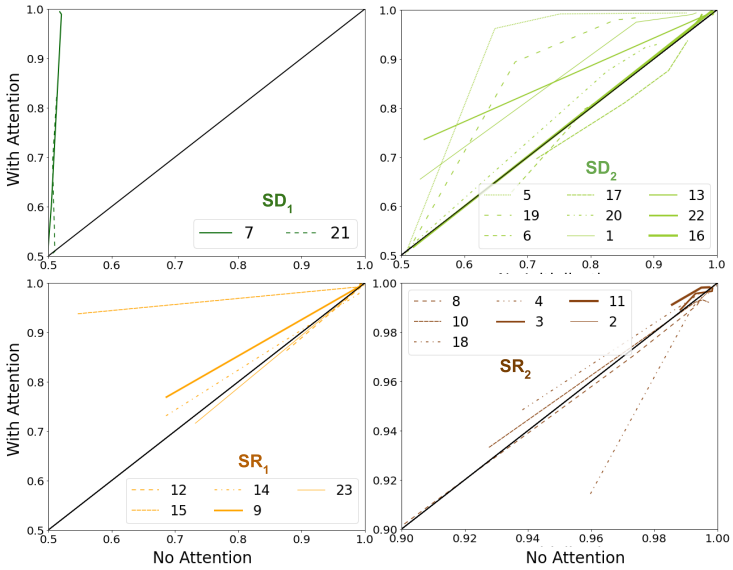
To measure the effectiveness of different forms of attention for solving the SVRT, we compared the accuracy of three ResNet-50 models: one capable of spatial attention, one capable of feature-based attention, and one that had no attention mechanisms ("vanilla") (Figure 3.2). Spatial attention consistently improved model accuracy on all tasks across all five dataset sizes that models we used for

training models. The improvement in accuracy is particularly noticeable for the SD_1 cluster. Tasks in this sub-cluster are composed of two rules, which ResNets, without attention, struggled to learn. Attention helps ResNets learn these tasks more efficiently. The improvement is also evident for SD_2 and SR_1. However, the benefit of attention for SR_2 is marginal since ResNets without attention already perform well on these tasks.

We find that feature-based attention leads to the largest improvements for SD_1, especially when training on 5k or 10k examples (Figure 3.3). On the other hand, spatial attention leads to the largest improvements for SD_2 and SR_1. This improvement is pronounced when training on 500 or 1000 examples. Taken together, the differential success of spatial versus feature-based attention reveals that their varying attentional demands can explain the task sub-clusters discovered in our data-driven taxonomy.



(a) Spatial attention



(b) Feature-based attention

Figure 3.2: Test accuracies for a baseline ResNet50 vs. the same architecture endowed with the two forms of attention for each of the twenty-three SVRT tasks when varying the number of training examples. A different axis scale is used for SR_2 to improve visibility. These curves are constructed by joining task accuracy for five points representing dataset sizes.

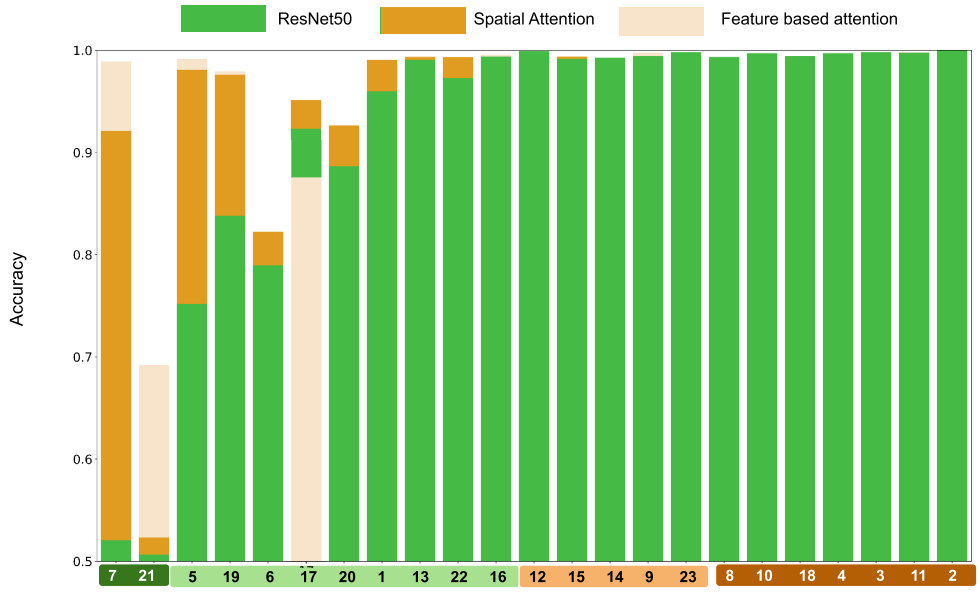


Figure 3.3: Test accuracies for 50-layer ResNets with spatial attention (orange), feature-based attention (tan), or no attention (green). Each bar depicts performance after training from scratch on 10k samples.

To better understand how the ResNet-derived taxonomy found in Experiment 1 can be explained by the need for spatial and feature-based attention, we measured the relative improvement of each form of attention over the vanilla ResNet. For each attention model and task, we calculated the ratio of the test accuracies between the model and the vanilla ResNet50. We repeated this for every training dataset size, then fit a linear model to these ratios to calculate the slope across dataset sizes (see Figure 3.4 for representative examples). We then repeated this procedure for all twenty-three tasks to produce two 23-dimensional vectors containing slopes for each model and every task.

We next used these slopes to understand the attentional demands of each SVRT task. We did this through a two-step procedure. First, we applied a principal component analysis (see Figure 3.5) to the vanilla ResNet performance feature vectors ($N = 15$) derived from Experiment 1. Second, we correlated the principal components with the slope vectors from the two attention models. We restricted our analysis to the first two principal components, which captured $\sim 93\%$ of the variance in the vanilla ResNet's performance (Figure 3.5). This analysis revealed a

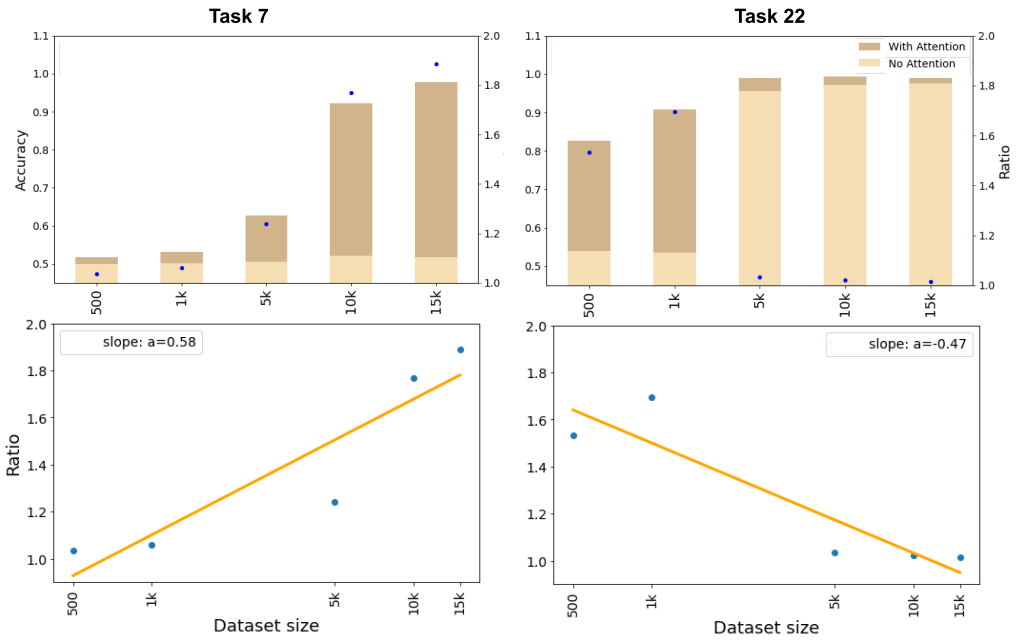


Figure 3.4: The benefit of attention in solving the SVRT is greatest in data-limited training regimes. The x-axis depicts the number of samples for training, and the y-axis depicts a ratio of the average performance of models with attention to models without attention. When the ratio is greater than 1, it shows that attention helps vs. hurts when lower than 1. This gives us five ratios per task and attention process corresponding to each dataset size. We performed a linear fitting procedure for these points and calculated the corresponding slope. This slope characterizes the relative benefits of attention for that particular task as the number of training examples available increases. If the benefit of attention is most evident in lower training regimes, one would expect a relatively small slope. If the benefit of attention is most evident in higher training regimes, one would expect a large slope.

dissociation between the two forms of attention: feature-based attention was most correlated with the first principal component, and spatial attention with the second principal component. Additionally, along the first principal component, we found the broader dichotomy of these 23 tasks into SD and SR clusters, whereas the second principal component divulges the tasks which responded better with spatial attention from tasks requiring either no attention or feature-based attention (as seen in dotted red line along both the axis in Figure 3.5). The corresponding Pearson coefficient r and p values are given in Table 3.1.

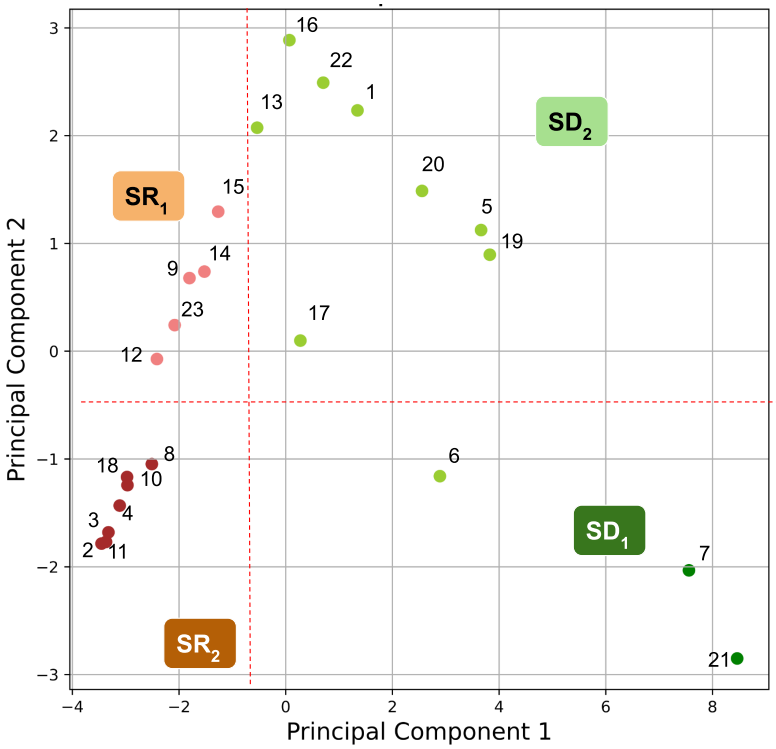


Figure 3.5: Principal component analysis of the twenty-three tasks using the 15-dimensional feature vectors derived from Experiment 1 representing the test accuracy obtained for each task for different dataset sizes and ResNets of varying depths (18, 50 & 152). The dotted red line represents 4 different bins in which these tasks can be clustered.

To summarize our results from Experiment 2, we have found that the task clusters derived from ResNet test accuracies computed over a range of depth and training set sizes can be explained in terms of attentional demands. Here, we have shown that endowing these networks with attentional mechanisms helps them learn some of the most challenging problems with far fewer training examples. We also found that the relative improvements obtained over standard ResNets with feature-based and spatial attention are consistent with the taxonomy of visual reasoning tasks found in Experiment 1. More generally, our analysis shows how the relative need for feature vs. spatial attention seems to account for a large fraction of the variance in computational demand required for these SVRT tasks

Table 3.1: Pearson coefficient (r) and corresponding p values obtained by correlating the slope vectors of the spatial attention and the feature-based attention modules with the two principal components of Figure 3.5. See text for details.

	Spatial		*Feature*	
	r	**p**	**r**	**p**
PC_1	0.466	0.0249	**0.649**	0.0008
PC_2	**-0.652**	0.0007	-0.491	0.0174

defined in Experiment 1 according to their learnability by ResNets.

3.3 Experiment 2: Feature vs. rule learning

The learnability of individual SVRT tasks reflects two components: the complexity of the task's visual features and, separately, the complexity of the rule needed to solve the task. To what extent are our estimates of learnability driven by either of these components? We tested this question by training a new set of ResNets without attention according to the procedure laid out in Experiment 1, but with different pre-training strategies. One of the ResNets was pre-trained to learn visual statistics (but not rules) of SVRT images, and another was pre-trained on ImageNet, [a popular computer vision dataset containing natural object categories; Deng et al., 2009].

For pre-training on SVRT, we sampled 5,000 class-balanced images from each of the 23 tasks (5,000 $\times$ 23 = 115,000 samples in total). To ensure the networks did not learn any of the SVRT task rules, we shuffled images and binary class labels across all twenty-three problems while pre-training the network. We then trained models with binary cross-entropy to detect positive examples *without discriminating tasks*. Our assumption is that shuffling images and labels removes any semantic information between individual images and SVRT rules. However, a network with sufficient capacity can still learn the corresponding mapping between arbitrary images and class labels (even though it cannot generalize it to novel samples). To learn this arbitrary mapping, the network has to be able to encode visual features; but by construction, it cannot learn the SVRT task rule. When training this model and the ImageNet-initialized model to solve individual

SVRT tasks, we froze the weights of the convolutional layers and only fine-tuned the classification layers to solve SVRT problems.

Figure 3.6 shows a comparison between the different architectures in terms of their test accuracies according to the sub-clusters discovered in Experiment 1. These results first confirm that the SVRT pre-training approach works because it consistently outperforms pre-training on ImageNet (Figure B5) or training from scratch. Interestingly, for the SR_2 sub-cluster, we found that the benefits of pre-training on SVRT go down very quickly as the number of training examples grows. We interpret these results as reflecting the fact that generic visual features are sufficient for the task and that the rule can be learned very quickly (somewhere around 500 and 5,000 samples). For SR_1 sub-cluster, the benefits of starting from features learned from SVRT are somewhat more evident in low training regimes. Still, these advantages quickly vanish as more training examples are available (the task is learned by all architectures within 5,000 training samples).

For SD_1 while there appears to be a noteworthy advantage of pre-training on SVRT over ImageNet pre-training and training from scratch, the tasks never appear to be fully learned by any of the networks even with 15,000 training examples. This demonstrates the challenge of learning the rules associated with this sub-cluster beyond simply learning good visual representations. Finally, our results also show that the performance gap across all the architectures for SD_2 vs. SD_1 increases rapidly with more training examples – demonstrating the fact that the abstract rule for SD_2 tasks are more rapidly learned than for SD_1.

Finally, we carried out a similar analysis with the pre-trained network as done in Experiment 2: We built test accuracy vectors for the SVRT pre-trained network trained using all five dataset sizes (.5k, 1k, 5k, 10k, 15k) and searching over a range of optimal learning rates (*1e-4, 1e-5, 1e-6*). This led to a five-dimensional vector, which we normalized by dividing each entry with the corresponding test accuracy of a baseline ResNet50 trained from scratch. Hence, the normalized vector represents the improvement (ratio larger than 1) or reduction in accuracy (ratio smaller than 1) that results from the pre-training on SVRT for that particular task and training set size. We then calculated the slope vector in $\mathcal{R}^{(23)}$, which we correlated with the corresponding spatial and feature-based attention vectors from Experiment 2.

We found that task improvements due to SVRT pre-training correlated more strongly with task improvements due to spatial ($r = 0.90$, $p = 4e-9$) than

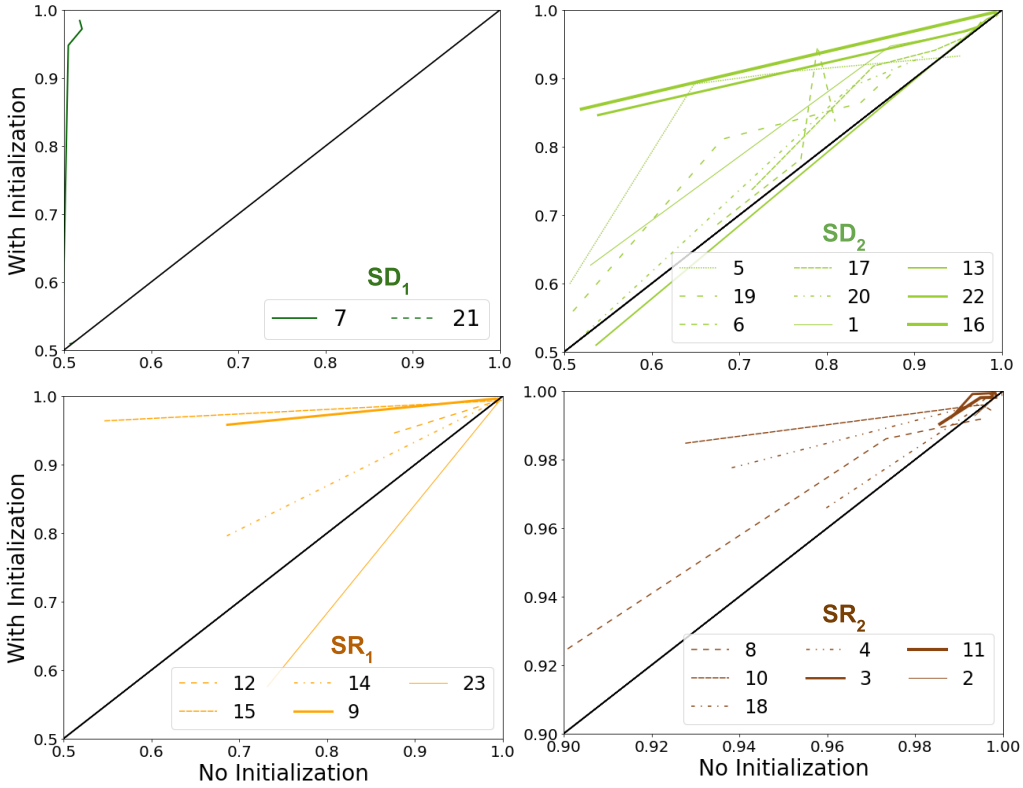


Figure 3.6: Test accuracies for a baseline ResNet50 trained from scratch ("No initialization") vs. the same architecture pre-trained on an auxiliary task in order to learn visual representations that are already adapted to the SVRT stimuli for different numbers of training examples. The format is the same as used in Figure 3.2. A different axis scale is used for SR_2 to improve visibility. These curves are constructed by joining task accuracy for five points representing dataset sizes.

feature-based attention ($r = 0.595$, $p = 0.002$). This suggests that the observed improvements in accuracy derived from spatial attention are more consistent with learning better feature representations compared to feature-based attention.

To summarize, in Experiment 3, we have tried to address the question of the learnability of SVRT features vs. rules. We found that using an auxiliary task to pre-train the networks on the SVRT stimuli in order to learn visual representations beforehand provides learning advantages to the network compared to a network trained from scratch.

We also found a noteworthy correlation between the test accuracy vector of a network pre-trained on SVRT visual statistics and a similar network endowed with spatial attention. This suggests that spatial attention helps discover the abstract rule more so that it helps improve learning good visual representations for the task.

3.4 Conclusion

Earlier, Kim et al. [2018] hypothesized that such straining by convolutional networks is due to their lack of attention mechanisms to allow the explicit binding of image regions to mental objects. A similar point was made by Greff et al. [2020] in the context of the contemporary neural network failure to carve out sensory information into discrete chunks which can then be individually analyzed and compared (see also Tsotsos et al. [2007] for a similar point). Interestingly, this prediction was recently tested using human EEG by Alamia et al. [2021a] who showed that indeed the brain activity recorded during SD tasks is compatible with greater attention and working memory demands than SR tasks. At the same time, that CNNs can learn SR tasks more efficiently than SD tasks does not necessarily mean that human participants can solve these tasks without attention. Indeed, [Logan, 1994b] has shown that SR tasks such as judging insideness require attention under some circumstances.

To assess the role of attention in visual reasoning, we used Transformer modules to endow deep CNNs with spatial and feature-based attention. The relative improvements obtained by the CNNs with the two forms of attention varied across tasks. Many tasks reflected a larger improvement for spatial attention, and a smaller number benefited from feature-based attention. Further, we found that the

patterns of relative improvements accounted for much of the variance in the space of SVRT tasks derived in Experiment 1. Overall, we found that the requirement for feature-based and spatial attention accounts well for the taxonomy of visual reasoning tasks identified in Experiment 1. Our computational analysis also lead to testable predictions for human experiments by suggesting tasks that either benefit from spatial attention (task *22*) or from feature-based attention (task *21*), tasks that benefit from either form of attention (task *19*), and tasks that do not benefit from attention (task *2*).

Finally, our study has focused on the computational benefits of spatial and feature-based attention for visual reasoning. Future work should consider the role of other forms of attention, including object-based attention [Egly et al., 1994b] for visual reasoning.

In our second experiment, we studied the learnability of SVRT features vs. rules. We did this by pre-training the neural networks on auxiliary tasks in order to learn SVRT features before training them to learn the abstract rules associated with individual SVRT problems. Our pre-training methods led to networks that learn to solve the SVRT problems better than networks trained from scratch as well as networks that were pre-trained to perform image categorization on the ImageNet dataset. We have also found that such attention processes seem to contribute more to rule learning than to feature learning. For SR_1 sub-cluster we find this type of pre-training to be advantageous in lower training regimes but the benefits rapidly fade away in higher training regimes. In contrast, this pre-training does not allow the tasks from the SD_1 sub-cluster to be learned even with 15k samples – suggesting that the key challenge with these tasks is not to discover good visual representations but rather to discover the rule. This suggests the need for additional mechanisms beyond those implemented in ResNets. This is also consistent with the improvements observed for these tasks with the addition of attention mechanisms.

In summary, our study compared the computational demands of different visual reasoning tasks. While our focus has been on understanding the computational benefits of attention and feature learning mechanisms, it is clear that additional mechanisms will be required to fully solve all SVRT tasks. These mechanisms are likely to include working memory which is known to play a role in SD tasks [Alamia et al., 2021a]. Overall, this work illustrates the potential benefits of incorporating brain-like mechanisms in modern neural networks and provides a path forward to achieving human-level visual reasoning.

Chapter 4

CHAPTER 4

ROLE OF SELF-ATTENTION IN A COGNITIVE ARCHITECTURE

4.1 Introduction 50
4.2 Related Work 51
4.3 Proposed approach 53
4.4 Hyperparameters 57
4.5 Method 59
4.6 Benchmarking guided attention 61
4.7 Benchmarking the system 63
4.8 Learning Compositionality 64
4.9 Zero-shot generalization 65
4.10 Ablation Study 66
4.11 Additional Experiment 67
4.12 Conclusion and limitations 69

Intelligence is not only the ability to reason; it is also the ability to find relevant material in memory and to deploy attention when needed.

– Daniel Kahneman

4.1 Introduction

Abstract reasoning refers to our ability to analyze information and discover rules to solve arbitrary tasks, and it is fundamental to general intelligence in human and non-human animals [Gentner and Markman, 1997, Lovett and Forbus, 2017]. It is considered a critical component for the development of artificial intelligence (AI) systems and has rapidly started to gain attention. A growing body of literature suggests that current neural architectures exhibit significant limitations in their ability to solve relatively simple visual cognitive tasks in comparison to humans (see Ricci et al. [2021] for review).

Given the vast superiority of animals over state-of-the-art AI systems, it makes sense to turn to brain sciences to find inspiration to leverage brain-like mechanisms to improve the ability of modern deep neural networks to solve complex visual reasoning tasks. Indeed, a recent human EEG study has shown that attention and memory processes are needed to solve same-different visual reasoning tasks [Alamia et al., 2021b].

It is thus not surprising that deep neural networks which lack attention and/or memory system fail to robustly solve visual reasoning problems that involve such same-different judgments [Kim et al., 2018]. Recent computer vision work [Messina et al., 2021b, Vaishnav et al., 2022a] has provided further computational evidence for the benefits of attention mechanisms in solving a variety of visual reasoning tasks. Interestingly, in both aforementioned studies, a transformer module was used to implement a form of attention known as self-attention [Cheng et al., 2016, Parikh et al., 2016]. In such a static module, attention mechanisms are deployed in parallel across an entire visual scene.

By contrast, modern cognitive theories of active vision postulate that the visual system explores the environment dynamically via sequences of attention shifts to select and route task-relevant information to memory [Ullman, 1984, 1987]. Psychophysics experiments [Hayhoe, 2000] on overt visual attention have shown that eye movement patterns are driven according to task-dependent routines.

Inspired by active vision theories, we describe a dynamic extension of the self-attention mechanisms popularised by the transformer module, which we call *guided attention*. Our proposed transformer-based Guided Attention Module for (visual) Reasoning (GAMR) learns to shift attention dynamically, in a task-dependent manner, based on queries internally generated by an LSTM executive controller. Through extensive experiments on the two main visual reasoning challenges, the Synthetic Visual Reasoning Test (SVRT) [Fleuret et al., 2011] and the Abstract Reasoning Task (ART) [Webb et al., 2021], we demonstrate that our neural architecture is capable of learning complex compositions of relational rules in a data-efficient manner and performs better than other state-of-the-art neural architectures for visual reasoning. Using explainability methods, we further characterize the visual strategies leveraged by the model in order to solve representative reasoning tasks. We demonstrate that our model is compositional – in that it is able to generalize to novel tasks efficiently and learn novel visual routines by recomposing previously learned elementary operations.

4.2 Related Work

Previous studies [Alamia et al., 2021b] have shown that mechanisms associated with attention and memory are involved in solving same-different visual reasoning tasks. Multiple datasets have been used to assess the visual reasoning ability of neural networks. One of the first challenges included the SVRT. Recently introduced Raven's Progressive Matrices (RPM) dataset [Barrett et al., 2018, Zhang et al., 2019] focuses on seven unique relations to be learned. However, it was found that the dataset was seriously flawed as it was later found that neural architectures could solve tasks by leveraging shortcuts [Hu et al., 2020, Spratley et al., 2020]. Prior work on SVRT studies has focused on the role of attention in solving some of these more challenging tasks. In SVRT, some of the tasks are significantly more challenging for computer vision algorithms than others. In particular, tasks that involve same-different (SD) judgements appear to be significantly harder for neural networks to learn compared to those involving spatial relation

judgement (SR) [Stabinger et al., 2016b, Yihe et al., 2019b, Kim et al., 2018] (see Ricci et al. [2021] for a recent review). Motivated by neuroscience principles, Vaishnav et al. [2022a] studied how the addition of feature-based and spatial attention mechanisms differentially affects the learnability of the tasks. These authors found that SVRT tasks could be further taxonomized according to their differential demands for these two types of attention. In another attempt to leverage a transformer network to incorporate attention mechanisms for visual reasoning, Messina et al. [2021b] proposed a recurrent extension of the classic Vision Transformer block (R-ViT). Spatial attention and feedback connections helped the transformer to learn visual relations better. The authors compared the accuracy of four same-different (SVRT) tasks (tasks *1,5,20,21*) to demonstrate the efficacy of their model. They also showed that, even with 400k samples available for training, neither a Relational Network [Santoro et al., 2017] nor a Vision Transformer [Dosovitskiy et al., 2021] were capable of learning these tasks. While a recent work Webb et al. [2021] has explored the role of memory in ART reasoning tasks.

With the introduction of transformer networks, attention mechanisms started gaining popularity in computer vision. They can either complement [Bello et al., 2019, Vaishnav et al., 2022a, d'Ascoli et al., 2021] or completely replace existing CNN architectures [Ramachandran et al., 2019, Touvron et al., 2021b, Dosovitskiy et al., 2021]. Augmenting the attention networks with the convolution architectures helps them explore the best of both and train relatively faster. In contrast, stand-alone attention architecture takes time to develop similar inductive biases as CNN. As initially introduced by Vaswani et al. [2017], transformer attention uses a key (k), query (q), and value (v) attention mechanisms for NLP. Since the images are not like language, all the existing architectures for image recognition explore self-attention variants in which the key, query, and values are the same as an input image. We used a similar attentional system, but instead of using it as a self-attention module, we called it a guided attention module. This system internally generates a query to guide the attention module to the location essential for the task. Since there could be more than one location where the model will attend, we then implemented a memory bank.

We took inspiration for the memory bank from Webb et al. [2021], where mechanisms for variable binding and indirection were introduced in architecture for visual reasoning with the help of external memory. Variable binding is the ability to bind two representations, and indirection is the mechanism involved in retrieving one representation to refer to the other. These authors also introduce

Temporal Context Normalization (TCN) [Webb et al., 2020], which is found beneficial for out-of-distribution generalization for relational reasoning tasks. However, the model exhibits significant limitations: It assumes an object-centric image representation whereby objects are presented individually in a sequence. We cannot evaluate such an architecture on the SVRT challenge because images in each task contain multiple objects which require individuation. There are also some relations, like "touching", which this individuation cannot represent (or any object-centric architecture). ESBN also lacks an attentional mechanism and works best in a scenario where hard attention at the pre-processing level helps to simplify the tasks. We tested this template-matching behavior of the architecture by training it in the presence of Gaussian noise. It led to a chance-level performance. Here, we build on this work and describe an end-to-end trainable model that learns to individuate task-relevant scenes and store their representations in memory to allow the judging of complex relations between these objects. Finally, our relational mechanism is inspired by the work in Santoro et al. [2017] that introduced a plug-and-play model for computing relations between object-like representations in a network.

4.3 Proposed approach

Our model can be divided into three components: an encoder, a controller, and a relational module (see Fig. 4.1 for an overview).

The **encoder module** includes a feature extraction block (f_e) for an image (x_{in}) which is composed of five convolutional blocks (Figure 4.2). The output of the module is denoted as $z_{img} \in \mathcal{R}^{(128,hw)}$ (with h height and w width). We applied Temporal Context Normalization (TCN) as done in [Webb et al., 2020]. TCN is a simple inductive bias implemented similar to batch normalization but applied at the task-relevant temporal window. Using TCN helps to preserve the relational information between the objects within the window, resulting in better learnability and generalization. TCN is applied over z_{img} before passing it to the controller for further processing.

After z_{img} is built, a *guided-attention* block routes visual information from relevant image location at each time step (t). This block builds on the now classic *multi-head attention* (MHA) transformer module used in Natural Language Processing [Vaswani et al., 2017], which differs substantially from the self-attention

Algorithm 1 Memory and Attention-based visual REasOning model (*GAMR*). (||) indicates the concatenation of two vectors, forming a new vector. {, } indicates the concatenation of a matrix and a vector, forming a matrix with one additional row.

$k_{r_{t=1}} \leftarrow 0$
$h_{t=1} \leftarrow 0$
$M_{t=1} \leftarrow \{\}$
$z_{img} \leftarrow f_e(x_{in})$
for `t in 1...T` **do**
  $out,\ g,\ query_t,\ h_t \leftarrow f_s(h_{t-1},\ k_{r_{t-1}})$
  $w_k \leftarrow (guided\ attention(z_{img},\ query_t)).sum(axis = 1)$
  $z_t \leftarrow (z_{img}\ *\ w_k).sum(axis = 1)$
  if t is 1 **then**
    $k_{r_t} \leftarrow 0$
  else
    $w_{k_t} \leftarrow w_k.sum(axis = 2)$
    $k_{r_t} \leftarrow g\ *\ (M_{t-1}\ *\ w_{k_t})$
  end if
  $M_t \leftarrow \{M_{t-1},\ z_t\}$
end for
$all_{obj} \leftarrow r_\theta(\sum_{i,j=1}^{T}(M_{v_i},\ M_{v_j}))$
$\hat{y} \leftarrow f_\phi(all_{obj}\ ||\ out)$

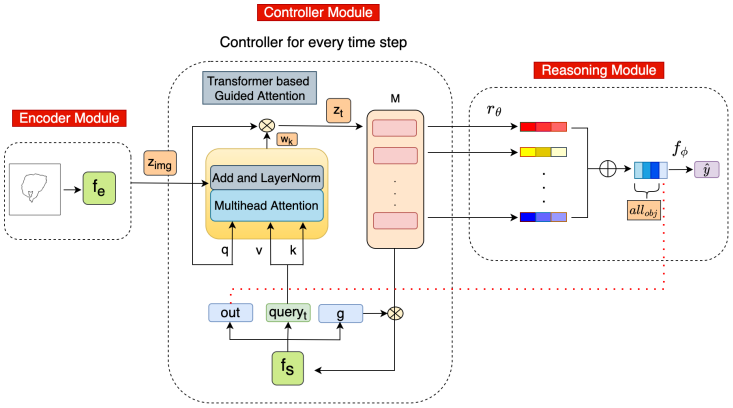


Figure 4.1: Our proposed *GAMR* architecture is composed of three components: an *encoder* module (f_e) builds a representation (z_{img}) of an image, a *controller* guides a transformer-based multi-head attention module to dynamically shift attention, and selectively route task-relevant object representations (z_t) to be stored in a memory bank (M). The recurrent controller (f_s) generates a query vector ($query_t$) at each time step to guide the next shift of attention based on the current fixation. After a few shifts of attention, a reasoning module (r_θ) learns to identify the relationships between objects stored in memory.

transformer module more commonly used in vision. An MHA block works as a retrieval block to extract relevant information based on 3 attention variables: key (k), query (q), and value (v). The query (q) is used to compute a similarity score with the key (k), which is then multiplied by the values (v). In our implementation, MHA accepts keys and values from a controller (f_s) (discussed in the next paragraph) and queries from the encoder block (f_e) to generate a weight vector ($w_k \in \mathcal{R}^{128}$) corresponding to the key (k). The vectors w_k is then used to modulate the feature vector z_{img} to yield a new context vector ($z_t \in \mathcal{R}^{128}$). A similar approach was used in Visual Question Answering (VQA) [Yu et al., 2019] to compute the

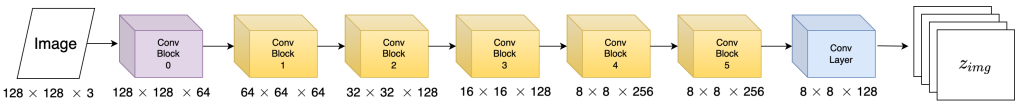


Figure 4.2: Encoder module (f_e) used in *GAMR*. It consists of four convolutional blocks to process input image of 128×128 resolution

similarities between questions and images. The context vector (z_t) is then stored in the memory bank (M) to be subsequently accessed again later by a reasoning module. This memory bank is inspired by the differential memory used in Webb et al. [2021].

The **controller module** is responsible for generating a query ($query_t$) in response to a task-specific goal in order to guide attention in the transformer module. The controller module (f_s) uses a Long Short-Term Memory (LSTM) to provide a query vector ($query_t \in \mathcal{R}^{128}$) as input to the guided attention module for the current time step t. The controller also generates a gate vector ($g \in \mathcal{R}^{128}$) and output vector ($out \in \mathcal{R}^{512}$). The gate vector g generates the next input to the controller based on prior relevant features stored in the memory. The gate (g) is later used to shift attention to the next task-relevant feature based on the features previously stored in M. On the other hand, the decision layer uses the output vector (out) to produce the system classification output (Figure 4.3).

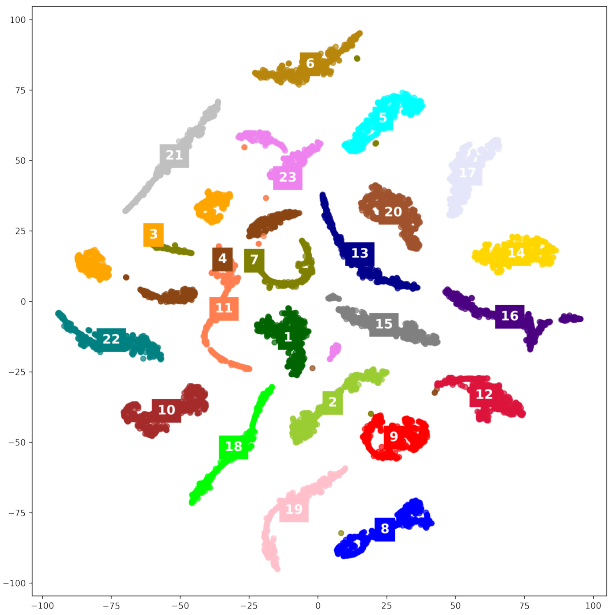


Figure 4.3: **Abstract variable:** t-SNE plot of the output vector (*out*) obtained from the controller (f_e) for all 23 SVRT tasks independently. Each cluster can be clearly identified from other clusters representing different relations learned. Tasks are represented as labels with the same colored box around them placed at the mean location of the cluster.

The **relational module** is where the reasoning takes place over the context vector (z_t) stored in the memory bank (M). This module is composed of a multilayer perceptron (MLP) layer (r_θ) which produces a relational vector (all_{obj}) similar to the relational network [Santoro et al., 2017]. As we will show in section 4.8, r_θ learns elementary operations associated with basic relational judgments between context vectors (z_t) stored in the memory (M). It is concatenated with the output (out) of the controller (f_s) at the last time step (t=T) and passed through decision layer (f_ϕ) to predict the output ($\hat{y}$) for a particular task. We have summarized the steps in Algorithm 1.

4.4 Hyperparameters

Number of heads : As the key and value vector for the Transformer based guided attention module has $\mathcal{R}^{128}$ dimension, changing the number of heads does not make any difference. However, we ran experiments on the same-different differentiation task of the SVRT Dataset. We evaluated *GAMR* with a varied number of heads in the multi-head attention module (no. of head = 1, 4, 8, 16). We found that head = 4 (Fig. 4.4) works best on average, and used this value for all our experiments. We have to investigate this further.

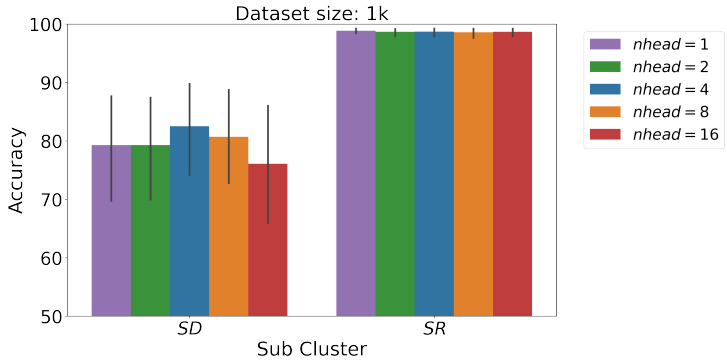


Figure 4.4: Ablation on Multi-Head Attention. We analyzed the average performance on SVRT tasks by changing the number of heads (*1, 2, 4, 8, 16*) and 1k images used for training. On average, this effect is distinguishable in SD tasks while SR tasks are already at their ceiling.

Table 4.1: **ART**: Number of training and test samples used for four different types of tasks.

Tasks		m=0	m=50	m=85	m=95
SD	Training	18,810	4,900	420	40
	Test	990	4,900	10,000	10,000
RMTS	Training	10,000	10,000	10,000	480
Dist3	Training	10,000	10,000	10,000	360
ID	Training	10,000	10,000	10,000	8,640
	Test	10,000	10,000	10,000	10,000

Holdout set For example, holdout *0* represents a generalization regime in which the test sets contain the same characters as those used during training. At the other extreme, in holdout *95*, the training set contains a minimal number of characters, most of which are actually used for tests. Hence, it is necessary to learn the abstract rule in order to generalize to characters in this regime.

SVRT This dataset can be generated with the code[1] provided by the SVRT authors with images of dimension 128×128. No augmentation technique was used for training other than normalization and randomly flipping the image horizontally or vertically, as is customary for this challenge [Vaishnav et al., 2022a].

We trained the model for a maximum of 100 epochs with a stopping criterion of 99% accuracy on the validation set. The model was trained using Adam [Kingma and Ba, 2014] optimizer and a binary cross-entropy loss. All the models were trained from scratch. We used a hyperparameter optimization framework *Optuna* [Akiba et al., 2019] to get the best learning rates, and weight decays for these tasks and reports the test accuracy for the models which gave the best validation scores.

[1] https://fleuret.org/cgi-bin/gitweb/gitweb.cgi?p=svrt.git;a=summary

Table 4.2: **ART**: For four different tasks number of epochs and learning rates (LR) used to train different architectures.

Tasks	m=0		m=50		m=85		m=95	
	GAMR							
	Epoch	LR	Epoch	LR	Epoch	LR	Epoch	LR
SD	50	0.0001	50	0.0005	100	0.0005	200	0.001
RMTS	50	0.00005	50	0.0001	50	0.0005	300	0.0005
Dist3	50	0.00005	50	0.0001	50	0.00005	300	0.0005
ID	50	0.00005	50	0.00005	50	0.0005	100	0.0005
	Other baselines							
SD	50	0.0005	50	0.0005	100	0.0005	200	0.0005
RMTS	50	0.0005	50	0.0005	50	0.0005	300	0.0005
Dist3	50	0.0005	50	0.0005	50	0.0005	300	0.0005
ID	50	0.0005	50	0.0005	50	0.0005	100	0.0005

Learning compositionality One of the triplets (*21, 19, 25*) involves rotation which needed more than 1,000 samples to learn. So, we selected 5,000 samples each from the tasks x and y to pre-train the network. This pre-training is carried out for 100 epochs for both tasks. Once the model is trained, we fine-tune it on the novel unseen task z. We confirmed that *GAMR* was able to learn the new rule with as few as ten samples per category – hence demonstrating an ability to harness compositionality.

4.5 Method

Datasets We used two datasets for our experiments, SVRT and ART. Experiments in section 4.6, 4.7, 4.8, 4.9 is carried out with SVRT dataset and in section 4.11 with ART dataset.

The SVRT dataset is composed of 23 different binary classification challenges, each representing either a single rule or a composition of multiple rules. A complete list of tasks with sample images from each category is shown in Figures A1, A2. We formed four different datasets with 0.5k, 1k, 5k, and 10k training samples to train our model. We used unique sets of 4k and 40k samples for validation and

test purposes. Classes are balanced for all the analyses.

We trained the model for a maximum of 100 epochs with a stopping criterion of 99% accuracy on the validation set. The model was trained using Adam [Kingma and Ba, 2014] optimizer and a binary cross-entropy loss. All the models were trained from scratch. We used a hyperparameter optimization framework *Optuna* [Akiba et al., 2019] to get the best learning rates, and weight decays for these tasks and reports the test accuracy for the models which gave the best validation scores.

Webb et al. [2021] proposed four visual reasoning tasks that we will henceforth refer to as the *Abstract Reasoning Task* (ART): (1) a same-different (SD) discrimination task, (2) a relation match to sample task ($RMTS$), (3) a distribution of three tasks ($Dist3$) and (4) an identity rule task (ID). These four tasks utilize shapes from a set of 100 unique Unicode character images [2]. They are divided into training and test sets into four generalization regimes using different holdout character sets (m = 0, 50, 85, and 95) from 100 characters.

Baselines For the baselines in this dataset, we compared our architecture performance to a Relational Network (RN), a popular architecture for reasoning in VQA. The RN uses the same CNN backbone as *GAMR* with feature maps of dimension $\mathcal{R}^{128,hw}$ where $h = 8$ and $w = 8$. We consider each spatial location of the encoded feature representation as an object (i.e., $N = 8 \times 8 = 64$ object representations). We computed all pairwise combinations between all 64 representations using a shared MLP between all the possible pairs (4096 pairs). These combinations are then averaged and processed through another MLP to compute a relational feature vector before the final prediction layer (f_ϕ). In a sense, *GAMR* is a special case of an RN network endowed with the ability to attend to a task-relevant subset (N = 4) of these representations with the help of a controller instead of exhaustively computing all 4,096 possible relations – thus reducing the computing and memory requirements of the architecture very significantly.

As an additional baseline model we used ResNet-50 [He et al., 2016] ($ResNet$) and its transformer-based self-attention network (*Attn-ResNet*) introduced in Vaishnav et al. [2022a]. These have been previously evaluated on SVRT tasks [Funke et al., 2021b, Vaishnav et al., 2022a, Messina et al., 2021c,b]. It serves as a powerful baseline because of more free parameters and a self-attention module to com-

[2] https://github.com/taylorwwebb/emergent_symbols

pare the proposed active attention component of *GAMR*. In our proposed method, the controller shifts attention heads sequentially to individual task-relevant locations against a standard self-attention module where all task-relevant locations are attended to simultaneously. We also evaluated ESBN [Webb et al., 2021] in which we used a similar encoder to that of *GAMR* and passed the images in sequential order with each shape as a single stimulus and the number of time steps as the number of shapes present in the SVRT task. In order to train these models we used images of dimension 128×128 for architectures such as RN, $ESBN$, *GAMR* and 256×256 for *ResNet, Attn-ResNet* (consistent with previous work).

4.6 Benchmarking guided attention

Transformer [Vaswani et al., 2017] architectures used self-attention mechanisms to draw global dependencies between input and output. In self-attention, input interacts with itself to estimate where more attention has to be paid with the help of key, query and value dot product. Self-attention is composed of three steps (i) dot product similarity score, (ii) normalizing scores to obtain weights, and (iii) re-weighting the original embeddings using weights.

$$self\ attention = Softmax(\frac{QK^T}{\sqrt{d}})V$$

where 'd' is the dimension of K and V. To calculate the Q, K and V from an input X, there are trainable weight matrices W_q, W_k and W_v.

$$\begin{aligned} Q &= XW_q \\ K &= XW_k \\ V &= XW_v \end{aligned}$$

However, in *guided-attention*, we take query as z_{img}, key and value vectors as $query_t$ from the controller without any trainable matrix. So the attention score becomes

$$guided\ attention\ (GA) = softmax(\frac{z_{img}\ .\ query_t^T}{\sqrt{128}})query_t$$

as the dimensionality of the vector is $\mathcal{R}^{128}$. This guided-attention (GA $\in \mathcal{R}^{hw \times 128}$) score is summed across the spatial dimension ($h \times w$) to re-weight the feature channels of the z_{img} and form z_t vector. This module acts like a feature-based attention module because of its ability to modulate the channels and only store the feature vector in the memory.

We evaluated our guided-attention module (*GAMR*) and compared it with alternative systems with comparable base architecture but endowed with self-attention (*With-SA*) or no attention and/or memory (*GAMR w/o Atn (RN)*) over 23 SVRT tasks. As a side note, GAMR-SA turns out to be similar to ARNe [Hahne et al., 2019] used for solving Raven's tasks. We found that, on average, our Guided Attention Model's relative performance is 11.1% better than its SA counterpart and 35.6% than a comparable system lacking attention (or memory) for SD tasks; similarly, relative improvements for SR tasks are 4.5% and 10.4%. It shows that GAMR is computationally efficient as it yields a higher performance for the same number (1k) of training samples. Results are shown in Figure 4.5.

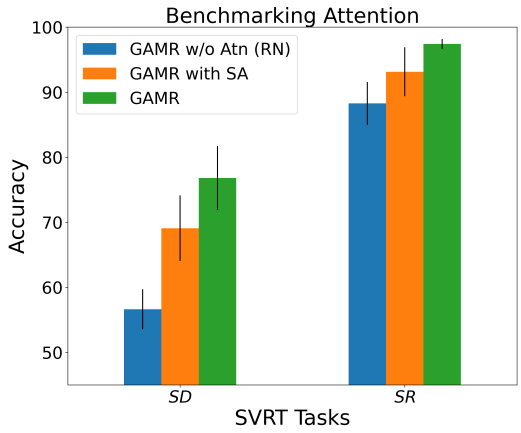


Figure 4.5: **Benchmarking Guided Attention**: We compared the average accuracy over two sub-clusters of SVRT obtained by *GAMR* with its variant when we replaced the guided-attention module with the self-attention (*GAMR-SA*) and when we completely gave away attention and made it a relational reasoning architecture (*GAMR w/o Atn (RN)*).

4.7 Benchmarking the system

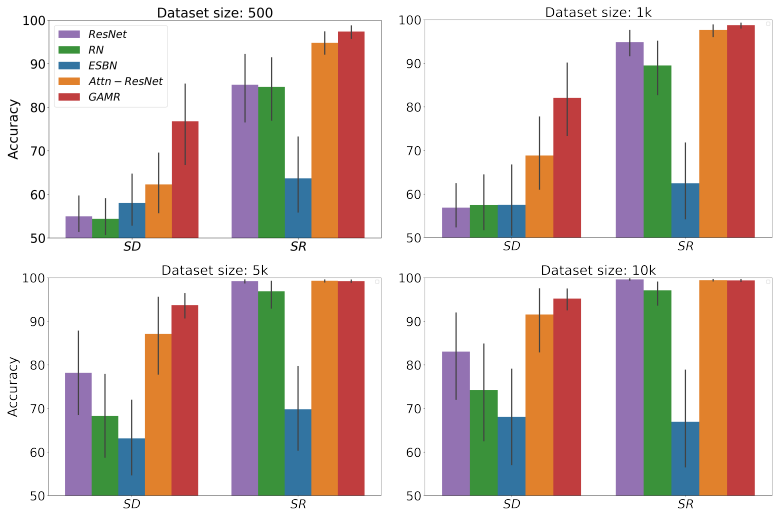


Figure 4.6: Bar plot analysis for the SVRT tasks grouped in same-different (SD) and spatially related (SR) tasks. We compared the accuracies of five baseline architectures with *GAMR*. ResNet-50 ($ResNet$) has 23M parameters, Relation Network (RN) has 5.1M parameters, ResNet-50 with attention (*Attn-ResNet*) has 24M parameters and *GAMR* & *ESBN* both have 6.6M parameters. We trained these with .5k, 1k, 5k and 10k samples.

All twenty-three tasks in the SVRT dataset can be broadly divided into two categories, same-different (SD) and spatial relations (SR), based on the identification of relations. Same-different (SD) tasks (*7, 21, 5, 19, 6, 17, 20, 1, 13, 22, 16*) have been found to be harder for neural networks [Ellis et al., 2015b, Kim et al., 2018, Stabinger et al., 2016b, 2021, Puebla and Bowers, 2021, Messina et al., 2021b, Vaishnav et al., 2022a] compared to spatial relations (SR) tasks (*12, 15, 14, 9, 23, 8, 10, 18, 4, 3, 11, 2*).

We analyzed an array of architectures and found that, on average, *GAMR* achieves at least 15% better test accuracy score on SD tasks for 500 samples. In contrast, for SR tasks, average accuracy has already reached perfection. We find a similar trend for other architectures when trained with different dataset sizes. Overall, RN (GAMR minus attention) and ESBN struggled to solve SVRT tasks even with 10k training samples, pointing towards the lack of an essential compo-

nent, such as attention. On the other hand, Attn-ResNet architecture demonstrated the second-best performance, which shows its importance in visual reasoning. Results are summarized in Figure 4.6.

4.8 Learning Compositionality

Below, we provide evidence that *GAMR* is capable of harnessing compositionality. We looked for triplets of tasks (x, y, z) such that z would be a composition of tasks x and y. We systematically looked for all such available triplets in the SVRT dataset and found three of them (*15, 1, 10*), (*18, 16, 10*) and (*21, 19, 25*). All these tasks and their associated triplets are described below. We study the ability of the network to learn to compose a new relation with very few training samples, given that it had previously learned the individual rules. We first trained the model with tasks x and y so that the rules are learned with the help of the reasoning module r_θ. The first layer learns the elementary operation over the context vectors stored in the memory block (M), and the second layer learns to combine those operations for the tasks z. We freeze the model after training with tasks x, y and only fine-tuned: (i) a layer to learn to combine elementary operations (r_θ) and (ii) a decision layer (f_ϕ) on tasks z with ten samples per category and 100 epochs total. Results are shown in Fig. 4.7.

We selected group corresponding to each tasks (*15, 18, 21*) used for composition. Task *15* has four shapes forming a square and are identical. It can be composed of task *1*, helping to identify the same shapes and task *10*, which helps to learn if the four shapes are forming a square. In task *18*, a rule is needed to be learned related to symmetry along the perpendicular bisector of the image. It can be taken as a composition of task *16* which requires learning mirror reflection of the image along the perpendicular bisector of the image and task *10* in which symmetry could be discovered in between 4 shapes (forming a square). At last, we took task *21*, which involves both scaling and rotation between two shapes in an image. As its compositional elements, we designed a variant where there is only rotation and no scaling and represented it with *25* and combined it with another counterpart of *21* where there is scaling and no rotation, i.e., task *19*.

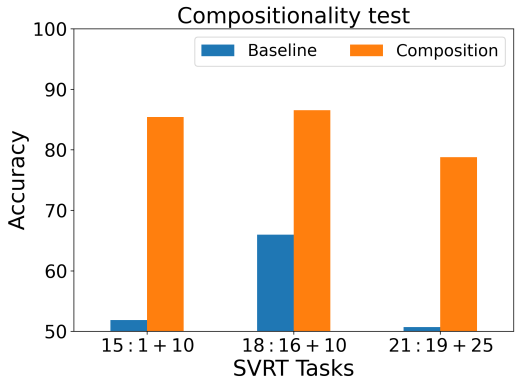


Figure 4.7: **Compositionality test**: We train the model with tasks containing specific rules (e.g., Task *1* representing same/different discrimination & task *10* involving identification if the four shapes form a square or not). We show that with its ability to compose already learned rules, *GAMR* can quickly learn with 10 samples per class to adapt to a novel scenario (e.g., *15* where the rule is to identify if the four shapes forming a square are identical or not.)

4.9 Zero-shot generalization

We hypothesize that if a model has learned the abstract rule underlying a given task, it should be able to re-use its knowledge of this task on other novel tasks which share a similar rule. To verify that *GAMR* is indeed able to generalize across tasks that share similar rules, we searched for pairs of tasks in SVRT which were composed of at least one common elementary relation [Vaishnav et al., 2022a] between them. For example, in pair (*1*, *22*), task *1* involves the identification of two similar shapes in category 1 and task *22* involves the identification of three similar shapes in category 1. In the selected pair, the category that judges the similar rule should belong to the same class (let us say category 1 in the above example) so that we test for the right learnability. We systematically identified a set x of tasks *1, 5, 7, 21, 23* representing elementary relations such as identifying same-different (*1, 5*), grouping (*7*), learning transformation like scaling and rotation (*21*) and learning insideness (*23*). Then we paired them with other tasks sharing similar relations. These pairs are task *1* with each of *5, 15 and 22*, task *5* with each of *1, 15 and 22*. Similarly other pairs of tasks are (*7, 22*), (*21, 15*) and (*23, 8*). We separately trained the model on the set x and tested the same model on

Training Task	Test Task	Test Accuracy		
		GAMR	Attn-ResNet	ResNet
1	5	72.07	53.03	**73.04**
	15	**92.53**	92.07	78.87
	22	**84.91**	80.10	67.15
5	1	**92.64**	85.73	92.28
	15	**84.36**	62.69	49.95
	22	**76.47**	55.69	50.19
7	22	**83.80**	79.11	50.37
21	15	**90.53**	50.00	49.76
23	8	**85.84**	58.90	59.25

Table 4.3: Test accuracy to show if the model learns the correct rules when we train it with a task and test on a different set of SVRT tasks with *GAMR*, Attention with ResNet50 (Attn-ResNet) and ResNet-50 (ResNet).

their respective pairs without finetuning further with any samples from the test set (zero-shot classification). We observed that *GAMR* could easily generalize from one task to another without re-training. On the contrary, a chance level accuracy by ResNet-50 (ResNet) shows the network's rote memorization of task-dependent features. In comparison, *GAMR* exhibits far greater abstraction abilities – demonstrating an ability to comprehend rules in unseen tasks without any training at all. Table 4.3 summarizes all the results.

4.10 Ablation Study

We now proceed to study what role different components of the proposed architecture play in *GAMR*'s ability to learn reasoning tasks. In our first set of experiments, we selected essential building blocks of the model, such as the relational vector (all_{obj}), feature channel gate vector (g), weighing factor (w_{k_t}) at time step t corresponding to how relevant are the elements stored in memory in order to get the next query, the role of the controller's output (out) in the final decision layer (f_ϕ) and temporal context normalization (tcn) on the encoded representation. We studied the effect of these components on SD and SR categories. Our lesioning study revealed that TCN plays a vital role in the model's reasoning capability even

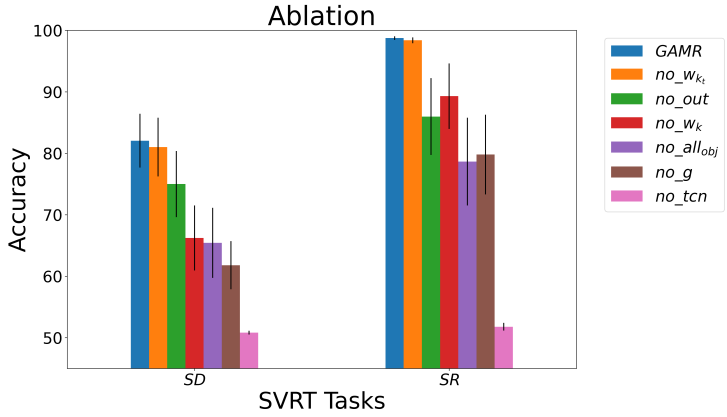


Figure 4.8: **Ablation studies**: We pruned separate parts of the model, one at a time, like the weighting factor at each time step (w_{k_t}), controller output (out), guided-attention (w_k), relational vector (all_{obj}), feature channel gain factor (g) and temporal context normalization (tcn) and show the variation in performance on SD and SR tasks when trained with 1k samples.

for learning simple rules, as in SR tasks. We also found that for SD tasks, excluding out from the decision-making process is detrimental. Finally, g and w_{k_t} have their visible role for tasks containing complex rules (task *7*, Figure 4.9). We have summarized the results in Figure 4.8. We also plot the saliency maps of the model in Figure 4.10 at each time step and show the way in which the model attends to task-dependent features while learning the rules.

4.11 Additional Experiment

Baseline models As a baseline, we chose the ESBN [Webb et al., 2021] along with the two other prevalent reasoning architectures, the Transformer [Vaswani et al., 2017] and Relation Network (RN) [Santoro et al., 2017]. These three share a similar encoder backbone as in *GAMR*. In order to make our baselines stronger, we evaluated these models in their natural order, i.e., by passing a single image at a time. We added a random translation (jittering) for the shapes in the area of ± 5 pixels around the center to prevent these architectures from performing template matching. For *GAMR*, we present task-relevant images together as a single stimu-

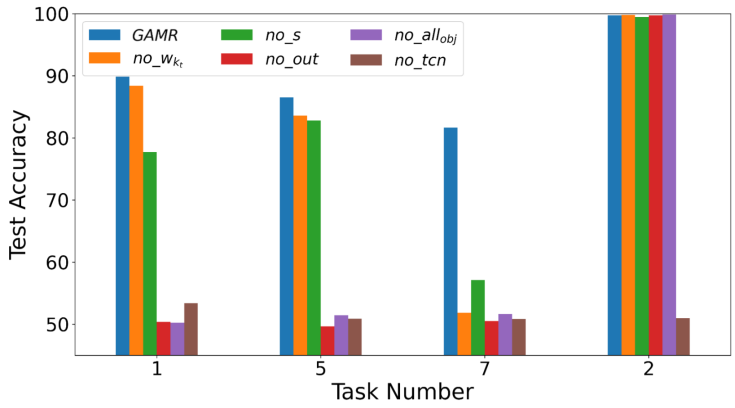


Figure 4.9: Ablation studies for *GAMR*. (a) We pruned separate parts of the model, one at a time, like the weighting factor at each time step (w_{k_t}), feature channel gain factor (g), controller output (out), relational vector (all_{obj}) and temporal context normalization (tcn) and show the variation in performance on tasks 1, 5, 7 and 2 when trained with 1k samples.

lus (Fig. 4.11) while jittering each shape. We have also added ART results where each image is centered and put together in a single stimulus in Figure 4.12. In order to make our architecture choose one option from multiple stimuli ($RMTS$: 2, $Dist3$ or ID: 4), we concatenate the relational vector (all_{obj}) for every stimulus and pass them to a linear layer for final decision.

Results We observed a near-chance level accuracy for all the baseline models and in different generalization scenarios for the SD and RMTS tasks (Figure 4.13). Whereas, when we trained the networks like ESBN with the images centered in the stimulus in a similar scenario, they resulted in perfect accuracy. However, our proposed architecture is robust to handle this jittering, as shown in Figure 4.12 where we compare its performance when images are not jittered. For the other two tasks, $Dist3$ and ID, baseline models performed better than the chance level (25%). ESBN showed an increasing trend in accuracy for progressively easier generalization conditions approaching 0 holdouts. This points toward the fact that the first three shapes in both these tasks allow ESBN to consider a translation factor while comparing the next three shapes, letting it choose the correct option

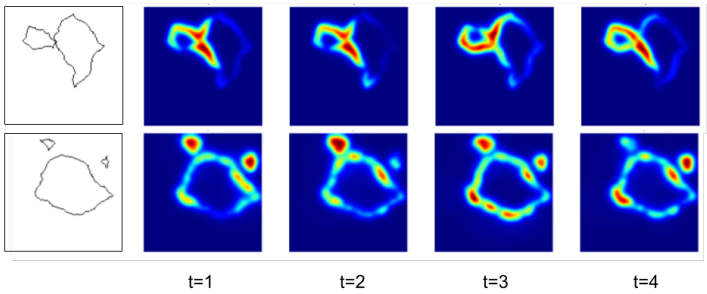


Figure 4.10: **Time steps**: Shift of attention with each time step in a task-dependent manner. In the first row, the task is to answer if the two shapes are touching each other from the outside. At each time step, the network explores the area where the shapes are touching each other. In the second row, tasks represented required to answer if one of the smaller shapes is inside a larger shape. The controller module for this task shifts attention across different shapes at each time step.

accordingly. RN and Transformer still struggled to generalize. ESBN (memory-based model) performance on SD tasks in both the visual reasoning dataset shows the importance of attention needed for reasoning.

4.12 Conclusion and limitations

In this paper, we described a novel transformer-based Guided Attention Module for (visual) Reasoning (*GAMR*) to try to start bridging the gap between the reasoning abilities of humans and machines. Inspired by the cognitive science literature, our module learns to dynamically allocate attention to task-relevant image locations and store relevant information in memory. Our proposed guided-attention mechanism is shown to outperform the self-attention mechanisms commonly used in vision transformers. Our ablation study demonstrated that an interplay between attention and memory was critical to achieving robust abstract visual reasoning. Furthermore, we demonstrated that the resulting systems efficiently are capable of solving novel tasks without limited training – by simply rearranging the elemental processing steps to learn the rules without involving any training. We demonstrated GAMR's versatility, robustness, and ability to generalize compositionality through an array of experiments. We achieved state-of-the-art accuracy for the

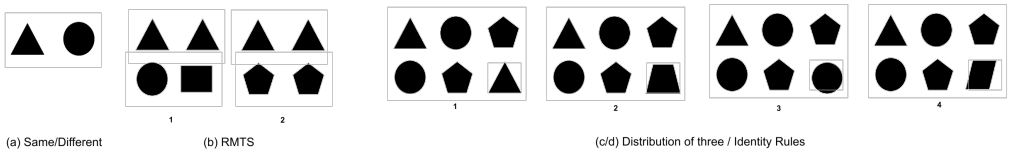


Figure 4.11: **ART for *GAMR*:** (a) Same/different discrimination task. (b) Relational match-to-sample task (answer is 2). (c) Distribution-of-three task (answer is 1). (d) Identity rules task (ABA pattern, answer is 3).

two main visual reasoning challenges in the process.

One limitation of the current approach is that it currently only deals with a fixed number of time steps (t=4). Training the model with four time-steps was sufficient to solve all SVRT and ART tasks efficiently. However, a more flexible approach is needed to allow the model to automatically allocate a number of time steps according to the computational demand of the task.

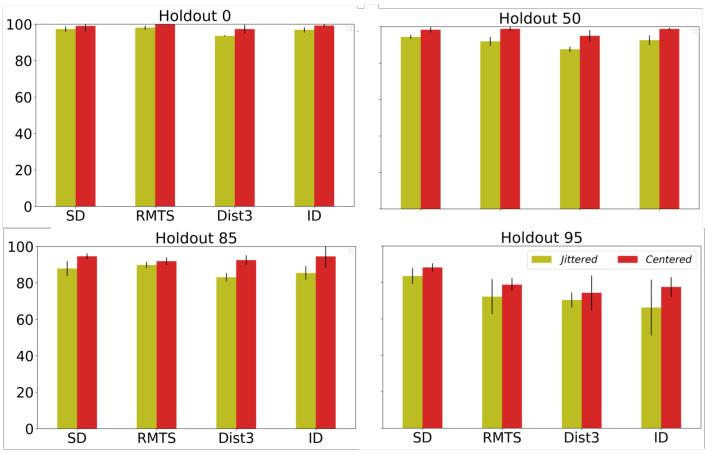


Figure 4.12: Test accuracy on ART with different holdout sets when the images are $centered$ and compare the accuracy when shapes are $jittered$ in every image. We find that unlike other baselines experiencing a huge drop in performance when shapes are jittered, GAMR is stable. We plot the average accuracy over ten runs on the dataset. x axis corresponds to the four types of tasks, and y represents the average accuracy score. These tasks are as follows: (a) same-different (SD) discrimination task, (b) Relation match to sample task (RMTS); (c) Distribution of three tasks (Dist3); and (d) Identity rule task (ID).

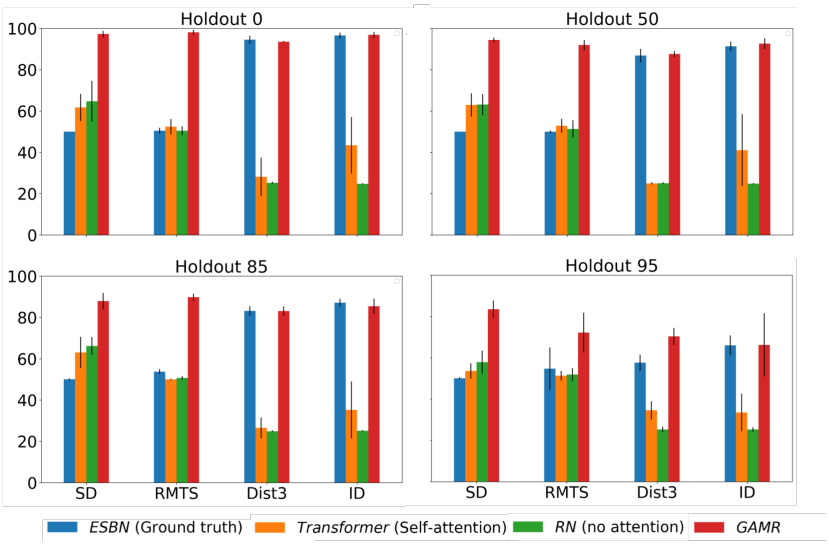


Figure 4.13: **ART:** Comparing the average performance of *GAMR* with other baselines over 10 runs for different holdout values (m = 0, 50, 85, 95). These models are evaluated on four types of tasks, i.e., Same-Different (SD), Relation match to sample (RMTS), Distribution of 3 (Dist3) and Identity rules (ID).

Chapter 5

CHAPTER 5

DISCUSSION AND FUTURE WORK

Attention is widely studied across several domains, including cognitive science and machine learning. It has deeply penetrated the field of computer vision and NLP, which has experienced a surge of self-attention-based architectures achieving state-of-the-art performance on numerous benchmarks. Furthermore, attention is a cognitive process providing the ability to concentrate on a relevant stimulus. This characteristic feature plays a vital role in enriching humans' reasoning ability. To better understand the self-attention mechanism, this thesis studied its role in cognitive and computer vision architectures under the purview of visual reasoning.

Visual reasoning is the process of analyzing the provided visual information in order to solve a task. It is considered an important part of fluid intelligence, which involves thinking and reasoning independent of learning, education, and experience. This ability has not only been shown in primates [Gentner et al., 2021b] but also in bees [Giurfa et al., 2001] and in newborn ducklings [Martinho and Kacelnik, 2016]. On the contrary, prior studies [Puebla and Bowers, 2021, Kim et al., 2018, Ricci et al., 2021, Messina et al., 2021c] (including our own work) have shown that modern-day neural networks struggle to solve simple visual reasoning tasks when tested on a popular benchmark called synthetic visual reasoning test (SVRT) by Fleuret et al. [2011] otherwise simple for humans. We found a similar trend when we tested popular reasoning architectures like Relational Network [Santoro et al., 2017], Transformer [Vaswani et al., 2017], ESBN [Webb et al., 2021] on Abstract Reasoning Task (ART) where the stimulus

contains a simple Unicode character. As a result, visual reasoning has become an increasingly popular topic of research in recent years with the emergence of numerous fluid intelligence tests for AI algorithms, including tests for Compositional Visual Reasoning (CVR) [Zerroug et al., 2022], Ravens' (RPM) [Barrett et al., 2018, Zhang et al., 2019] and visual progressive matrices (V-PROM) [Barrett et al., 2018, Teney et al., 2020] as well as an Abstract Reasoning Corpus (ARC) [Chollet, 2019].

We began this thesis by studying the computational mechanisms involved in solving the Synthetic Visual Reasoning Test (SVRT) challenge [Fleuret et al., 2011]. This challenge consists of twenty-three binary classification tasks, each involving unique abstract relations in their formulation. Previous studies have identified two broad categories of SVRT tasks [Stabinger et al., 2016a, Kim et al., 2018, Yihe et al., 2019a] – tasks involving spatial-relation (*SR*) judgment and tasks involving same-different (*SD*) judgment. The same-different tasks are found to be harder for the neural networks compared to the spatial relation tasks [Ellis et al., 2015b, Kim et al., 2018, Stabinger et al., 2016b, 2021, Puebla and Bowers, 2021, Messina et al., 2021b, Vaishnav et al., 2022a]. Consistent with this work, we proposed a novel taxonomy beyond the two primary clusters, reflecting the number of relationships used to define a particular task. A closer examination is needed to better understand the trend reflected by the neural networks in terms of accuracy and the number of relations involved in defining a particular task. An earlier study by Kim et al. [2018] has also reported that feedforward neural networks demonstrate a 'straining' effect in solving tasks involving same-different relations and hypothesized that the straining effect might be because of the lack of attention. The same was also shown with a human EEG experiment by Alamia et al. [2021b] where higher activity is recorded in the lower β band while solving the same-different judgment when compared to spatial relation judgment indicating higher demands for attention and/or working memory. To test the same, in the next chapter, we focused on understanding the role of attention in solving visual reasoning tasks.

Inspired by the two types of visual attention, we proposed a self-attention module that can be used as a *feature-based* or *spatial* attention to augment the features of a feedforward network (ResNet50 [He et al., 2016]). We evaluated both types of attention-augmented neural networks on SVRT tasks and found that our proposed attentional models could solve the most challenging SVRT tasks efficiently. The relative improvements obtained by feedforward networks endowed

with the two different forms of attention varied across SVRT tasks. We observed that many tasks benefited from spatial attention mechanisms, whereas a few tasks from feature-based attention and showed a significant improvement. Our computational analysis also leads to testable predictions for human experiments by suggesting tasks that benefit from spatial attention (task 22) or feature-based attention (task 21), tasks that benefit from either form of attention (task 19), and tasks that do not benefit from attention (task 2). While we evaluated two types of attention systems, there is a future possibility to add experiments with the third type of attention – object-based attention [Duncan, 1984, Egly et al., 1994a, Vecera and Farah, 1994, Kramer et al., 1997]. Object-based attention focuses on the particular object rather than its spatial location or corresponding features.

In the last part of the thesis, we proposed a novel architecture, the Guided Attention Model for (visual) Reasoning (*GAMR*). We integrated both cognitive abilities humans use – attention and memory in solving reasoning tasks. It draws inspiration from the cognitive science literature on active vision, where the spotlight of attention is routed in the visual system to gather task-relevant information. According to the theory of active vision, the visual world is explored using rapid eye movements guided by shifts of visual attention. We designed a controller akin to the mechanisms involved in the active vision framework to route the spotlight of attention and send the task-relevant representations in the memory block later used for reasoning. In GAMR, the controller is implemented with the key/query/value-based self-attention layer. Contrary to the existing method where key, query and value vectors all correspond to the same vector, the query is internally generated at each time step in our model. It helps the controller to shift the spotlight of attention. One of the limitations of the current approach is the fixed number of time steps. I believe that a future continuation of this work could be to incorporate a mechanism to adapt the number of time steps based on the complexity of the task. For now, we have set the number of time steps as four for all the tasks; however, a simple task might require fewer time steps to arrive at a decision with high confidence. To make the model adaptive to the situation, one possibility could be to train it with a confidence variable as a stopping criterion.

While we have limited our analysis to synthetic visual reasoning datasets, a future possibility exists to test the models on a real-world dataset like V-PROM. It consists of images organized in a Ravens' style of reasoning with some context images and some choice images from which the correct answer is selected. Another possible direction is to think of an architecture that considers two important

traits – efficient use of data and efficient use of the computational resource. One way to design this architecture is by incorporating a read-and-write mechanism similar to a Neural Turing Machine [Graves et al., 2014]. Both these mechanisms will help the network read the already stored relations from memory and write them into the memory if they are novel. We expect such cognitive architecture to demonstrate higher-order reasoning ability, continual learning, compositionally, and meta-learnability.

We also evaluated ViT [Dosovitskiy et al., 2021] – a full self-attention architecture on SVRT tasks and found that it struggles to learn the simplest of the SVRT tasks; however, Messina et al. [2021b] conducted a similar study on a smaller subset of four SVRT tasks trained on 28k samples and found that a recurrent version of ViT – an attentional network with a convolutional backbone can learn those tasks. Adding convolutions in the early layers of ViT is found to help obtain better accuracy and improve sensitivity to the optimization settings [Xiao et al., 2021]. This observation motivated us to propose *Conviformer* [Vaishnav et al., 2022b] for another collaborative project on leaf-fossil classification. We propose a network to incorporate a convolutional network as the front end for a full self-attention-based vision transformer network enhancing its ability to process higher-resolution images. While bigger images hold great importance in computer vision applications like object detection, segmentation and fine-grained classification, they cannot be used with vision transformers because of the associated computational memory demand. *Conviformer* improves the performance of vision transformers by incorporating local features and infusing convolutional priors in a transformer architecture. We would like to see how convolution-induced vision transformers perform on SVRT tasks.

Concept learning is yet another exciting direction of research. One of the key features of human intelligence is the ability to quickly learn new concepts and use them to generalize to a novel scenario. A *concept* can be an idea representing a class of events (e.g., walking), objects (e.g., cats), or their properties (e.g., blue color). To test the concept learning ability of neural networks in a few-shot manner, we recently introduced a novel visual reasoning dataset, Compositional Visual Reasoning (*CVR*) [Zerroug et al., 2022]. This dataset is based on the principle of odd-one-out reasoning. In this form of reasoning task, three out of four samples follow a similar concept (rule) in their formulation, while the fourth does not. Each sample contains shapes similar to the shapes used in the SVRT challenge. It extends the variety of relations used in the formulation compared to previously

defined datasets like SVRT or RPM. We have also included compositionality prior in the dataset, where some elementary relations are used to compose the several tasks. The motivation is to push the community to build a compositional and sample-efficient network.

In this thesis, we made one of the very first attempts to explore self-attention from a visual reasoning perspective. Attention plays a crucial role in demonstrating visual reasoning abilities, and a better attentional model is expected to be better at reasoning. We showed how self-attention operations could be used as a computational model of a visual attention system representing spatial and feature-based attention and also as a model for active vision. While we found that self-attention is as effective in solving reasoning tasks as in other vision-related challenges, there is a need for additional analysis to figure out the fundamental mechanisms in a full self-attention model that restricts its sample-efficient learnability for reasoning tasks. Overall this work demonstrates the potential benefits of adding self-attention mechanisms in cognitive and computer vision architecture for solving visual reasoning tasks.

Chapter 6

CHAPTER 6

PUBLICATIONS

- **Mohit Vaishnav**, Remi Cadene, Andrea Alamia, Drew Linsley, Rufin VanRullen, Thomas Serre; "Understanding the Computational Demands Underlying Visual Reasoning." *Neural Computation* 2022; 34 (5): 1075–1099. doi: https://doi.org/10.1162/neco_a_01485
- **Mohit Vaishnav**, Thomas Serre. "GAMR: A Guided Attention Model for (visual) Reasoning." *ArXiv* abs/2206.04928 (2022)
- Aimen Zerroug, **Mohit Vaishnav**, Julien Colin, Sebastian Musslick, Thomas Serre. " A Benchmark for Compositional Visual Reasoning." *In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* abs/2206.05379 (2022)
- **Mohit Vaishnav**, Thomas Fel, Ivan Rodriguez, Thomas Serre. "Conviformers: Convolutionally guided Vision Transformer." *ArXiv* abs/2208.08900 (2022)

Appendix

APPENDIX A

SYNTHETIC VISUAL REASONING TASK

Table A.1: Each cell represents attempts participants took to solve seven consecutive correct categorizations. Here, row and column represents $task\ number$ and $participant\ number$. Entries containing "X" indicate that the participant failed to solve the problem, and those cells are not included in the marginal means. [Fleuret et al., 2011]

Task No.	Participant No.																				Mean	Fail
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
1	1	12	1	2	8	8	1	1	X	1	14	1	4	1	1	1	2	1	1	1	3.26	1
2	3	1	2	2	10	19	4	4	14	3	2	3	21	1	1	5	3	2	22	9	6.55	0
3	7	1	3	1	4	3	1	1	7	1	6	1	1	1	4	1	1	1	4	2	2.55	0
4	1	6	7	1	1	3	1	1	1	1	3	1	1	2	1	1	7	5	7	1	2.6	0
5	7	X	1	21	8	3	1	5	X	1	X	9	13	1	6	2	X	8	1	7	5.88	4
6	X	20	X	X	27	25	12	26	X	X	3	X	X	X	4	16	X	X	X	X	16.63	12
7	1	X	1	X	13	8	4	14	X	3	8	12	7	X	1	6	1	1	14	9	6.44	4
8	7	6	1	14	4	14	1	5	1	4	8	1	1	1	13	5	3	7	4	1	5.05	0
9	4	24	1	16	3	1	1	13	X	X	4	6	X	2	7	1	3	1	5	1	5.47	3
10	1	8	2	2	4	1	3	5	X	4	1	2	16	4	4	2	1	1	4	3	3.58	1
11	4	2	3	1	3	1	4	8	1	2	1	1	1	1	1	5	2	1	1	1	2.2	0
12	1	2	8	1	9	4	8	4	1	7	25	2	5	2	X	2	5	X	4	1	5.06	2
13	1	20	5	14	X	3	1	13	7	10	1	13	9	5	X	3	3	2	X	1	6.53	3
14	4	4	1	1	3	10	2	X	12	14	1	19	1	3	1	1	4	8	1	2	4.84	1
15	1	X	1	2	2	1	1	1	X	5	1	2	4	1	1	18	10	3	2	1	3.17	2
16	12	18	7	X	X	2	2	14	X	X	28	9	13	X	22	10	X	X	X	X	12.45	9
17	14	X	6	5	2	X	21	X	X	22	X	14	X	X	X	X	13	8	28	1	12.18	9
18	5	17	2	X	27	5	5	1	X	2	X	7	19	4	1	1	5	1	1	2	6.18	3
19	2	10	1	11	1	3	5	11	8	2	4	2	17	1	4	4	1	6	1	X	4.95	1
20	14	7	4	5	1	8	3	1	X	18	9	16	3	1	6	1	2	1	15	1	6.11	1
21	6	X	1	X	1	X	23	X	X	21	28	7	26	7	15	2	17	X	16	X	13.08	7
22	1	9	14	1	1	4	1	5	21	2	1	2	5	1	6	1	4	1	1	6	4.35	0
23	1	1	7	22	1	1	2	1	6	21	2	5	4	6	4	3	1	1	6	8	5.15	0
Mean	4.45	9.33	3.59	6.78	6.33	6.05	4.65	6.7	7.18	7.2	7.5	6.14	8.55	2.37	5.15	4.14	4.4	3.11	6.9	3.05		
No of Fails	1	5	1	5	2	2	0	3	12	3	3	1	3	4	3	1	3	4	3	4		

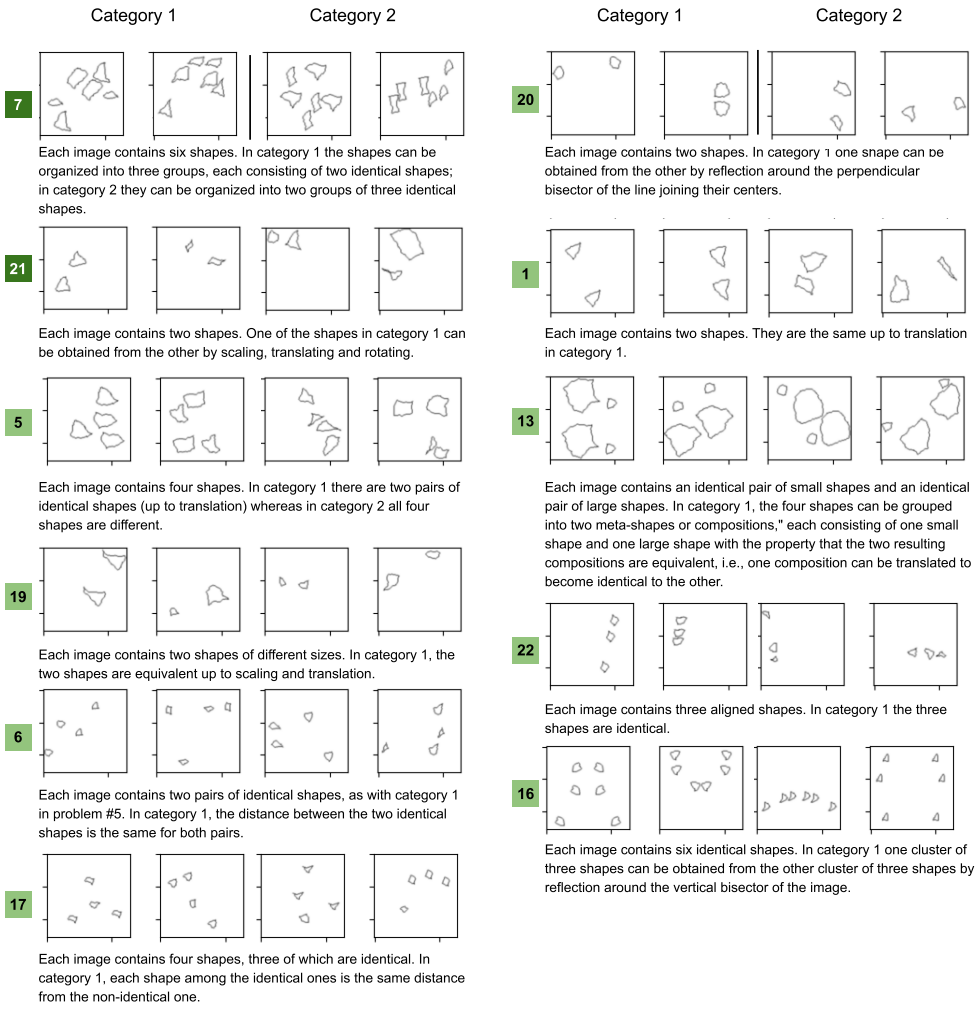


Figure A1: Sample images for Same Different (SD) tasks

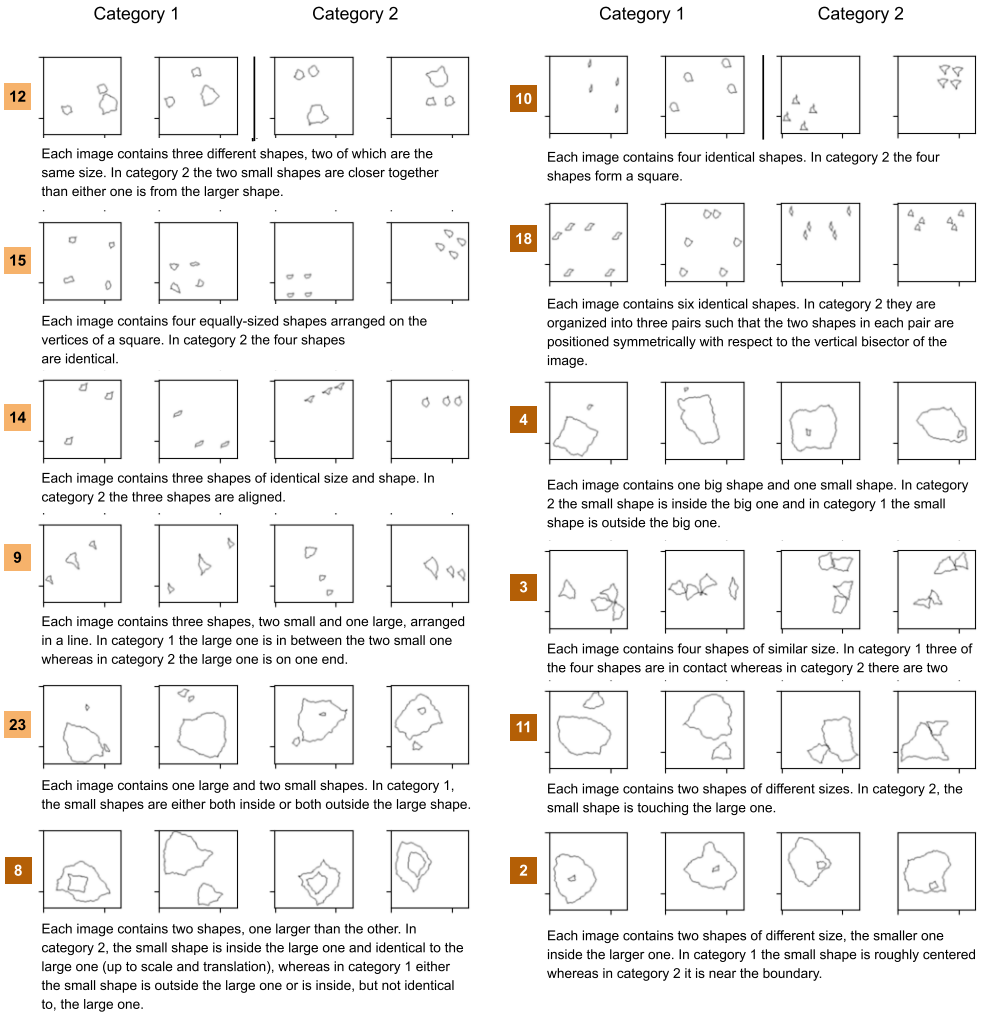


Figure A2: Sample images for Spatial Relation (SR) tasks

APPENDIX B

COMPUTATIONAL DEMANDS OF VISUAL REASONING

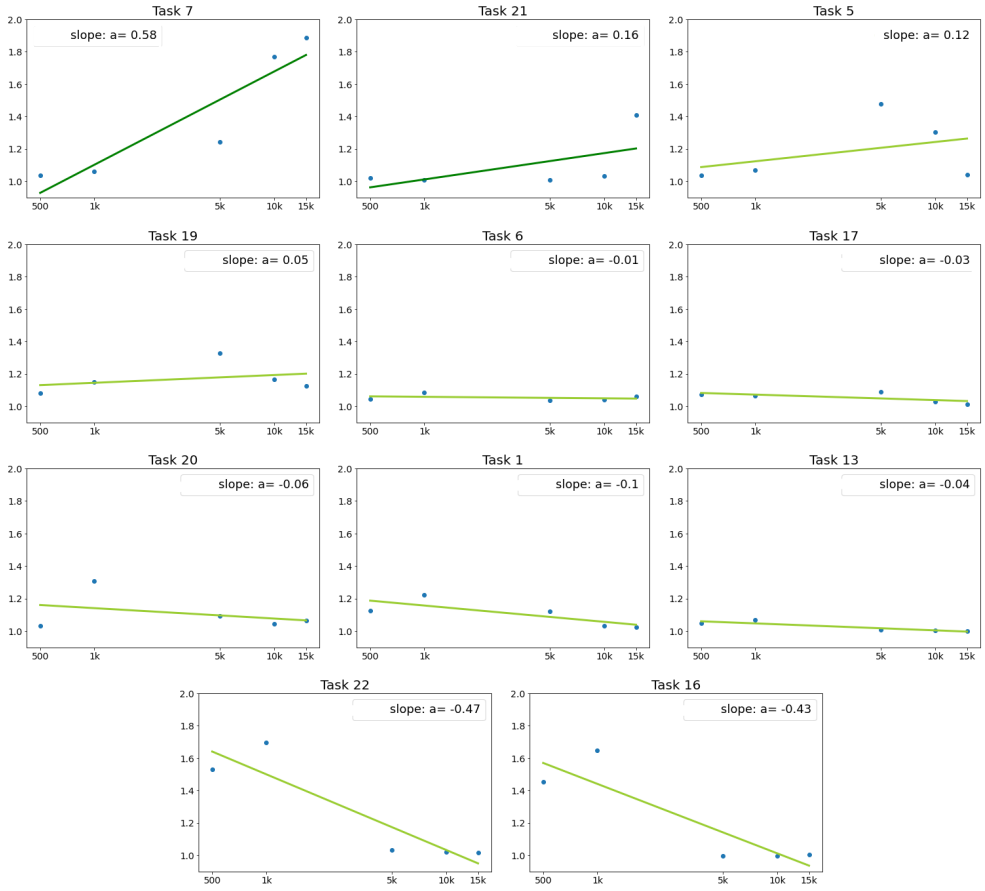


Figure B1: Slope attained by linear fitting of points obtained after taking the ratio of each of the network with spatial attention module and the test accuracy of a ResNet50 for each task and training condition for Same Different (SD) tasks

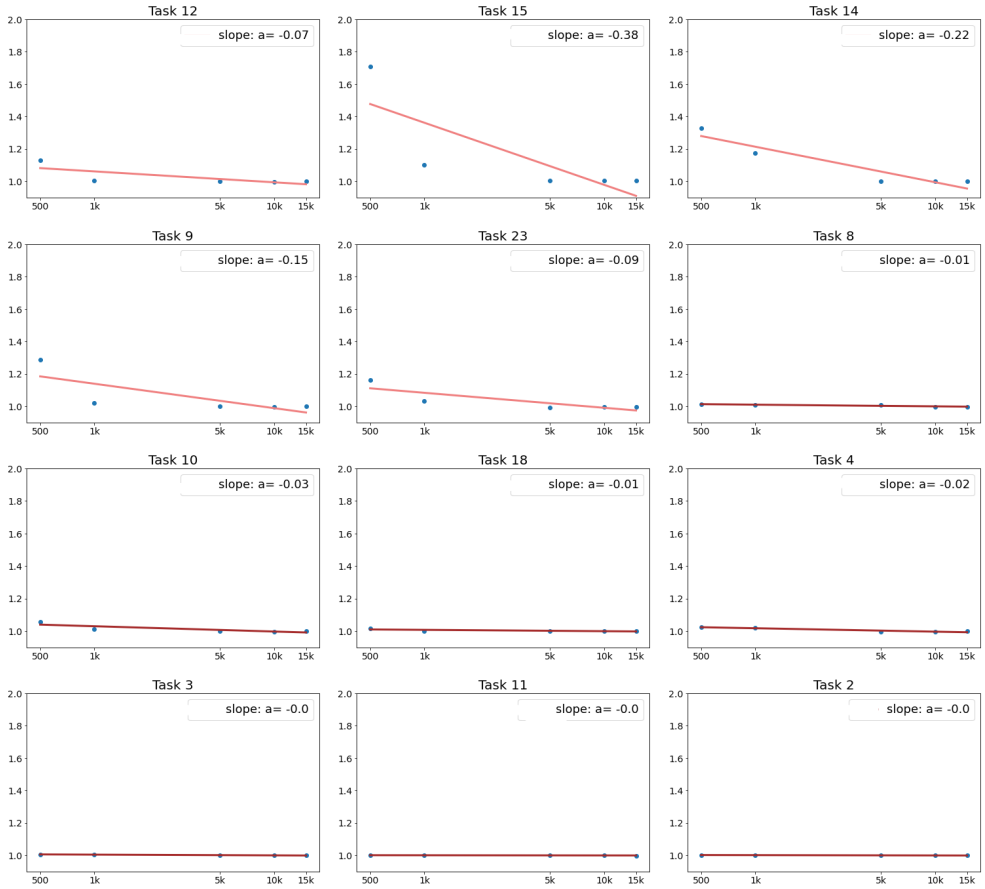


Figure B2: Slope attained by linear fitting of points obtained after taking the ratio of each of the network with spatial attention module and the test accuracy of a ResNet50 for each task and training condition for Spatial Relation (SR) tasks

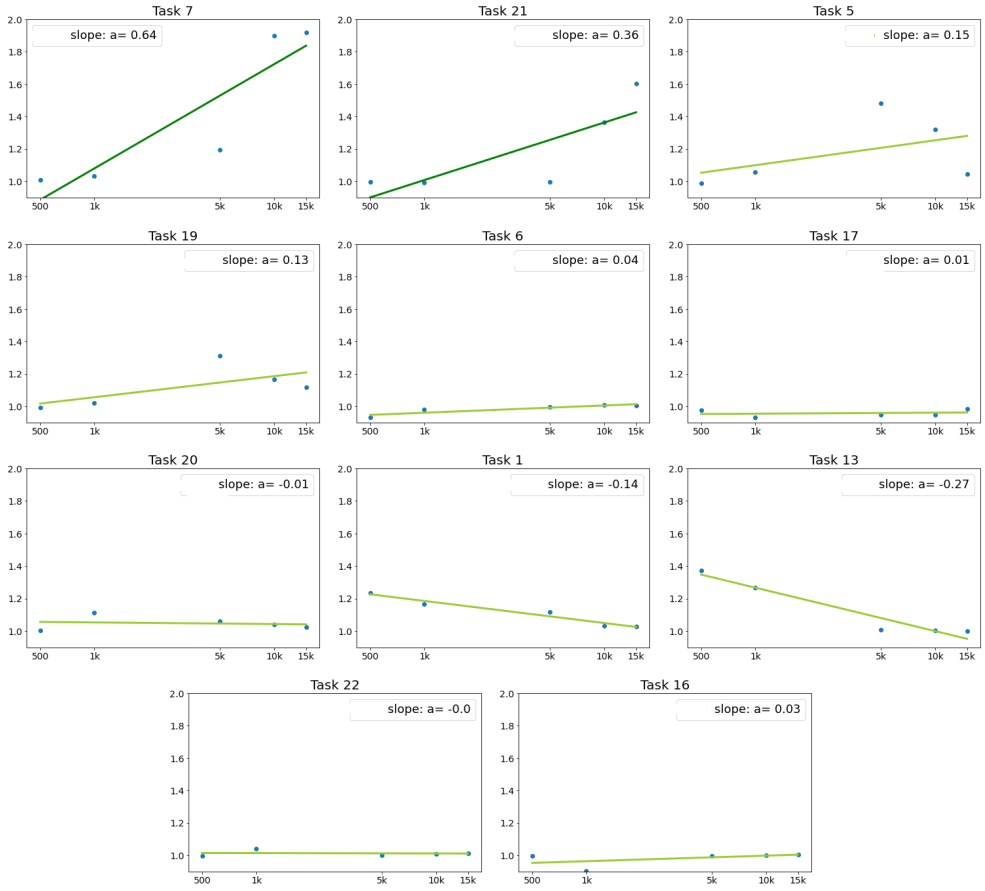


Figure B3: Slope attained by linear fitting of points obtained after taking the ratio of each of the network with feature-based attention module and the test accuracy of a ResNet50 for each task and training condition for Same Different (SD) tasks

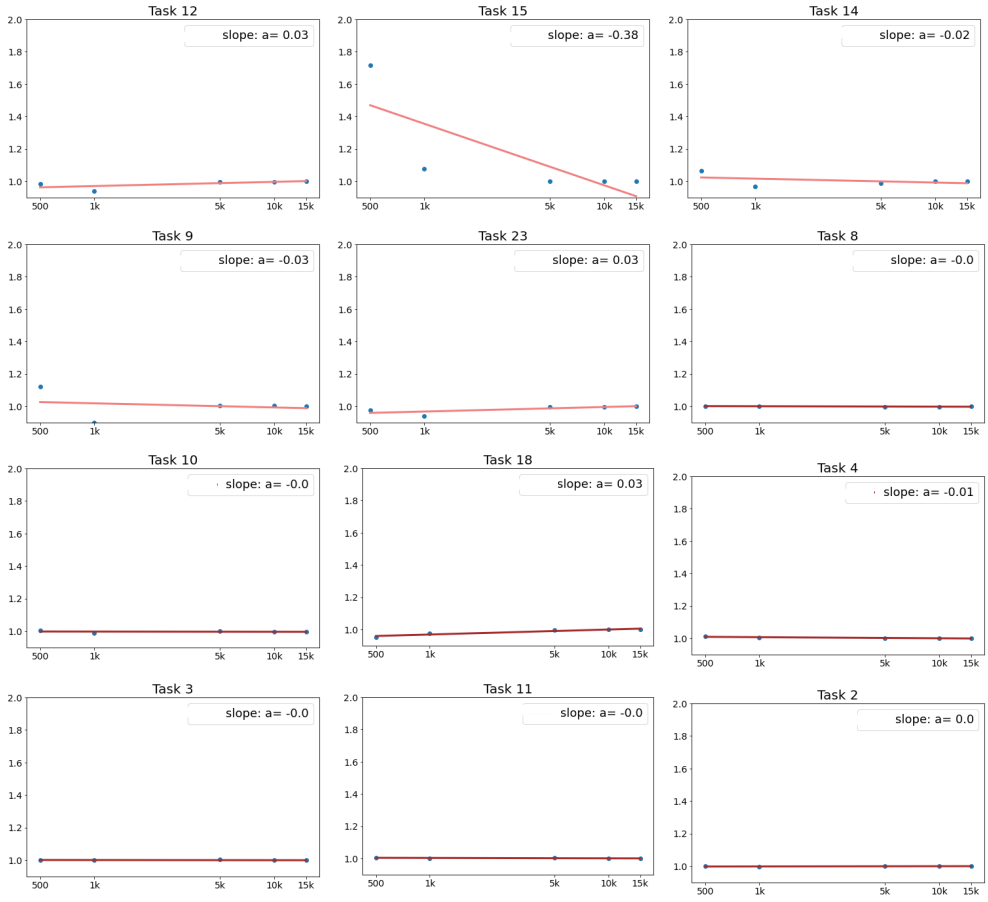


Figure B4: Slope attained by linear fitting of points obtained after taking the ratio of each of the network with feature-based attention module and the test accuracy of a ResNet50 for each task and training condition for Spatial Relation (SR) tasks

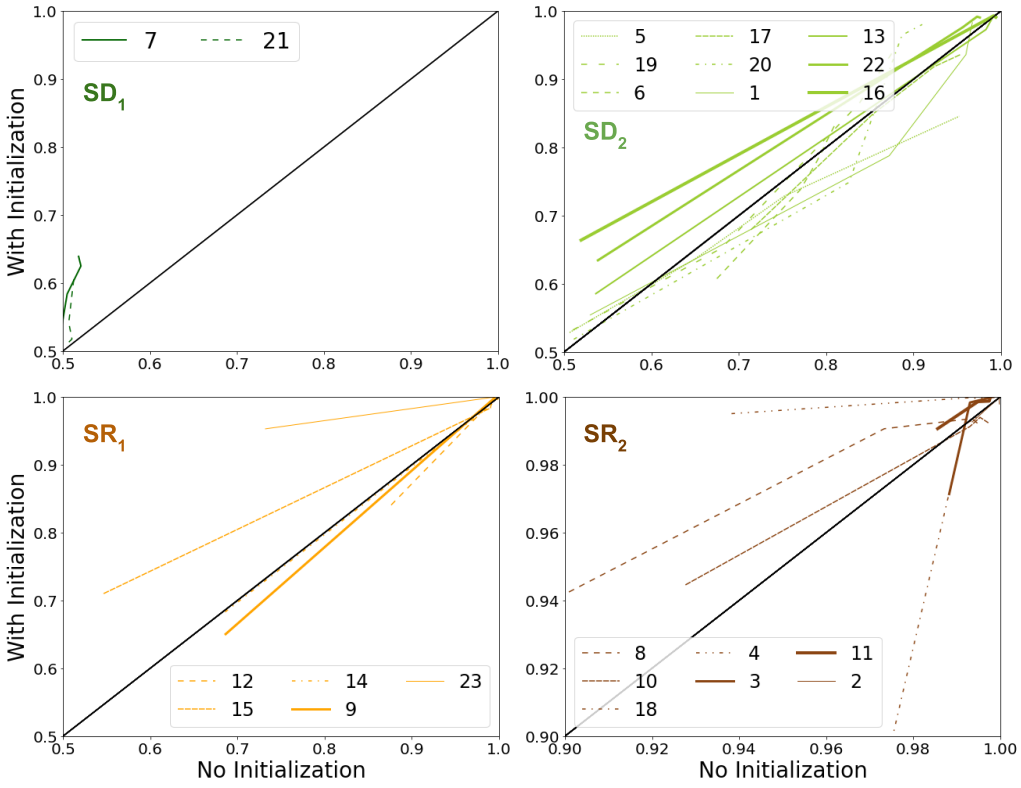


Figure B5: Test accuracies for a baseline ResNet50 trained from scratch ("No initialization") vs. the same architecture pre-trained on Imagenet data for different number of training examples. Also note that a different axis scale is used for SR_2 to improve visibility.

References

REFERENCES

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. (cit. on pp. ix, xvi, 6, 7, 8, 9, 11, 19, 35, 36, 52, 53, 61, 67, 74)

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy. (cit. on pp. ix, 10, 14, 15, 30, 35, 52, 77)

François Fleuret, Ting Li, Charles Dubout, Emma K Wampler, Steven Yantis, and Donald Geman. Comparing machines and humans on a visual categorization test. *Proceedings of the National Academy of Sciences*, 108(43): 17621–17625, 2011. (cit. on pp. xvi, 19, 22, 24, 25, 27, 29, 30, 51, 74, 75, 83)

Marvin M Chun, Julie D Golomb, Nicholas B Turk-Browne, et al. A taxonomy of external and internal attention. *Annual review of psychology*, 62(1):73–101, 2011. (cit. on pp. 3)

Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015. (cit. on pp. 3)

Subutai Ahmad. Visit: A neural model of covert visual attention. *Advances in neural information processing systems*, 4, 1991. (cit. on pp. 3)

Michael I Posner and Steven E Petersen. The attention system of the human brain. *Annual review of neuroscience*, 13(1):25–42, 1990. (cit. on pp. 3)

Claus Bundesen. A theory of visual attention. *Psychological review*, 97(4):523, 1990. (cit. on pp. 3)

Robert Desimone, John Duncan, et al. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995. (cit. on pp. 3, 35)

Maurizio Corbetta and Gordon L Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience*, 3(3):201–215, 2002. (cit. on pp. 3)

Steven E Petersen and Michael I Posner. The attention system of the human brain: 20 years after. *Annual review of neuroscience*, 35:73, 2012. (cit. on pp. 3)

Deepak Khosla, Christopher K Moore, David Huber, and Suhas Chelian. Bio-inspired visual attention and object recognition. In *Intelligent Computing: Theory and Applications V*, volume 6560, pages 17–27. SPIE, 2007. (cit. on pp. 3)

Grace W Lindsay and Kenneth D Miller. How biological attention mechanisms improve task performance in a large-scale visual system model. *ELife*, 7: e38105, 2018. (cit. on pp. 3)

Mohit Vaishnav, Remi Cadene, Andrea Alamia, Drew Linsley, Rufin VanRullen, and Thomas Serre. Understanding the Computational Demands Underlying Visual Reasoning. *Neural Computation*, pages 1–25, 02 2022a. ISSN 0899-7667. doi: 10.1162/neco_a_01485. URL https://doi.org/10.1162/neco_a_01485. (cit. on pp. 3, 12, 13, 50, 52, 58, 60, 63, 65, 75)

Mohit Vaishnav and Thomas Serre. Gamr: A guided attention model for (visual) reasoning. *arXiv preprint arXiv:2206.04928*, 2022. (cit. on pp. 3, 14)

Michael I Posner. Orienting of attention. *Quarterly journal of experimental psychology*, 32(1):3–25, 1980. (cit. on pp. 3)

Michael I Posner, Yoram Cohen, and Robert D Rafal. Neural systems control of spatial orienting. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 298(1089):187–198, 1982. (cit. on pp. 3)

John Duncan. Selective attention and the organization of visual information. *Journal of experimental psychology: General*, 113(4):501, 1984. (cit. on pp. 3, 76)

Robert Egly, Jon Driver, and Robert D Rafal. Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects. *Journal of Experimental Psychology: General*, 123(2):161, 1994a. (cit. on pp. 3, 76)

Shaun P Vecera and Martha J Farah. Does visual attention select objects or locations? *Journal of Experimental Psychology: General*, 123(2):146, 1994. (cit. on pp. 3, 76)

Arthur F Kramer, Timothy A Weber, and Stephen E Watson. Object-based attentional selection—grouped arrays or spatially invariant representations?: Comment on vecera and farah (1994). 1997. (cit. on pp. 3, 76)

Lisbeth Harms and Claus Bundesen. Color segregation and selective attention in a nonsearch task. *Perception & Psychophysics*, 33(1):11–19, 1983. (cit. on pp. 3)

Jon Driver and Gordon C Baylis. Movement and visual attention: the spotlight metaphor breaks down. *Journal of Experimental Psychology: Human perception and performance*, 15(3):448, 1989. (cit. on pp. 3)

Arthur F Kramer and Andrew Jacobson. Perceptual organization and focused attention: The role of objects and proximity in visual processing. *Perception & psychophysics*, 50(3):267–284, 1991. (cit. on pp. 3)

Gordon C Baylis and Jon Driver. Visual parsing and response competition: The effect of grouping factors. *Perception & Psychophysics*, 51(2):145–162, 1992. (cit. on pp. 3)

John Duncan and Ian Nimmo-Smith. Objects and attributes in divided attention: Surface and boundary systems. *Perception & Psychophysics*, 58(7):1076–1084, 1996. (cit. on pp. 3)

Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001. (cit. on pp. 4)

Melissa Saenz, Giedrius T Buracas, and Geoffrey M Boynton. Global effects of feature-based attention in human visual cortex. *Nature neuroscience*, 5(7): 631–632, 2002. (cit. on pp. 4)

Zhe Chen. Object-based attention: A tutorial review. *Attention, Perception, & Psychophysics*, 74(5):784–802, 2012. (cit. on pp. 4)

Victor AF Lamme and Pieter R Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in neurosciences*, 23(11): 571–579, 2000. (cit. on pp. 4)

Charles E Connor, Howard E Egeth, and Steven Yantis. Visual attention: bottom-up versus top-down. *Current biology*, 14(19):R850–R852, 2004. (cit. on pp. 4)

Timothy J Buschman and Earl K Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820): 1860–1862, 2007. (cit. on pp. 4)

David Soto, John Hodsoll, Pia Rotshtein, and Glyn W Humphreys. Automatic guidance of attention from working memory. *Trends in cognitive sciences*, 12(9):342–348, 2008. (cit. on pp. 5)

Christian NL Olivers and Martin Eimer. On the difference between working memory and attentional set. *Neuropsychologia*, 49(6):1553–1558, 2011. (cit. on pp. 5)

Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, April 1980. doi: 10.1007/bf00344251. URL https://doi.org/10.1007/bf00344251. (cit. on pp. 5)

Kunihiko Fukushima. Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics*, 26(23):4985–4992, 1987. (cit. on pp. 5)

Kunihiko Fukushima and Taro Imagawa. Recognition and segmentation of connected characters with selective attention. *Neural Networks*, 6(1):33–41, January 1993. doi: 10.1016/s0893-6080(05)80071-1. URL https://doi.org/10.1016/s0893-6080(05)80071-1. (cit. on pp. 5)

Eric O. Postma, H.Jaap van den Herik, and Patrick T.W. Hudson. SCAN: A scalable model of attentional selection. *Neural Networks*, 10(6):993–1015, August 1997. doi: 10.1016/s0893-6080(97)00034-8. URL https://doi.org/10.1016/s0893-6080(97)00034-8. (cit. on pp. 5)

Jürgen Schmidhuber and Rudolf Huber. Learning to generate artificial fovea trajectories for target detection. *Int. J. Neural Syst.*, 2:125–134, 1991. (cit. on pp. 6)

Milanese, Wechsler, Gill, Bost, and Pun. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 781–785, Los Alamitos, CA, USA, jun 1994. IEEE Computer Society. doi: 10.1109/CVPR.1994.323898. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.1994.323898. (cit. on pp. 6)

Florence Miau and Laurent Itti. A neural model combining attentional orienting to object recognition: Preliminary explorations on the interplay between where and what. In *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 789–792. IEEE, 2001. (cit. on pp. 6)

Albert Ali Salah, Ethem Alpaydin, and Lale Akarun. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):420–425, 2002. (cit. on pp. 6)

Dirk Walther, Laurent Itti, Maximilian Riesenhuber, Tomaso Poggio, and Christof Koch. Attentional selection for object recognition—a gentle way. In *International workshop on biologically motivated computer vision*, pages 472–479. Springer, 2002. (cit. on pp. 6)

Kerstin Schill, Elisabeth Umkehrer, Stephan Beinlich, Gerhard Krieger, and Christoph Zetzsche. Scene analysis with saccadic eye movements: top-down and bottom-up modeling. *Journal of electronic imaging*, 10(1):152–160, 2001. (cit. on pp. 6)

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. September 2014a. (cit. on pp. 6)

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166. (cit. on pp. 6)

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. (cit. on pp. 6)

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997a. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735. (cit. on pp. 6)

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://aclanthology.org/W14-4012. (cit. on pp. 6)

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1zJ-v5xl. (cit. on pp. 6)

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179. (cit. on pp. 6)

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article].

IEEE Computational Intelligence Magazine, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738. (cit. on pp. 7)

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014b. (cit. on pp. 7)

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. (cit. on pp. 7)

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. (cit. on pp. 7, 77)

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016. (cit. on pp. 7, 50)

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997b. (cit. on pp. 7)

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL https://aclanthology.org/D16-1244. (cit. on pp. 7, 50)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL http://arxiv.org/abs/1810.04805. (cit. on pp. 10)

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. (cit. on pp. 10)

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. (cit. on pp. 10)

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019. (cit. on pp. 10, 12, 13, 52)

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019. (cit. on pp. 10, 13, 52)

Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021a. (cit. on pp. 10)

Philipp Dufter, Martin Schmitt, and Hinrich Schütze. Position information in transformers: An overview. *Computational Linguistics*, pages 1–31, 2021. (cit. on pp. 10)

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018. (cit. on pp. 10)

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019. (cit. on pp. 10)

Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. (cit. on pp. 11)

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. (cit. on pp. 11, 12)

Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. (cit. on pp. 11)

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and

Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 10–15 Jul 2018a. URL https://proceedings.mlr.press/v80/parmar18a.html. (cit. on pp. 11)

Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12894–12904, 2021. (cit. on pp. 12)

Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16519–16529, 2021. (cit. on pp. 12)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. (cit. on pp. 12, 24, 60, 75)

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. (cit. on pp. 12, 13, 35)

Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. *Advances in neural information processing systems*, 31, 2018. (cit. on pp. 12)

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. (cit. on pp. 12, 35)

Kaiyu Yue, Ming Sun, Yuchen Yuan, Feng Zhou, Errui Ding, and Fuxin Xu. Compact generalized non-local network. *Advances in neural information processing systems*, 31, 2018. (cit. on pp. 12)

Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 3531–3539, January 2021. (cit. on pp. 12)

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019. (cit. on pp. 13)

James C. R. Whittington, Joseph Warren, and Tim E.J. Behrens. Relating transformers to models and neural representations of the hippocampal formation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=B8DVo9B1YE0. (cit. on pp. 13)

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020. (cit. on pp. 13)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b. (cit. on pp. 14)

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. 14, 43)

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021a. (cit. on pp. 15, 30)

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021b. (cit. on pp. 15)

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12175–12185, 2022. (cit. on pp. 15)

Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021. (cit. on pp. 15)

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12259–12269, 2021. (cit. on pp. 15)

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. (cit. on pp. 15)

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 367–376, October 2021. (cit. on pp. 15)

Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. (cit. on pp. 15)

Jingkai Zhou, Pichao Wang, Fan Wang, Qiong Liu, Hao Li, and Rong Jin. Elsa: Enhanced local self-attention for vision transformer. *arXiv preprint arXiv:2112.12786*, 2021. (cit. on pp. 15)

Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021. (cit. on pp. 16, 52)

Mohit Vaishnav, Thomas Fel, Ivan Felipe Rodrıguez, and Thomas Serre. Conviformers: Convolutionally guided vision transformer. *arXiv preprint arXiv:2208.08900*, 2022b. (cit. on pp. 16, 18, 77)

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. (cit. on pp. 16)

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018b. (cit. on pp. 16)

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. (cit. on pp. 16)

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13693–13696, April 2020. doi: 10.1609/aaai.v34i09.7123. URL https://doi.org/10.1609/aaai.v34i09.7123. (cit. on pp. 18)

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Toward more efficient training of deep networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxsrgStvr. (cit. on pp. 18)

Taylor Whittington Webb, Ishan Sinha, and Jonathan Cohen. Emergent symbols through binding in external memory. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LSFCEb3GYU7. (cit. on pp. 19, 51, 52, 56, 60, 61, 67, 74)

Li Fei-Fei, F F Li, A Iyer, C Koch, and P Perona. What do we perceive in a glance of a real-world scene? *J. Vis.*, 7(1):1–29, 2007. (cit. on pp. 22)

Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. *Proc. Natl. Acad. Sci. U. S. A.*, 112 (12):3618–3623, 2015. (cit. on pp. 22)

Gabriel Kreiman and Thomas Serre. Beyond the feedforward sweep: feedback computations in the visual cortex. *Ann. N. Y. Acad. Sci.*, February 2020. (cit. on pp. 22)

Gordon D Logan. *On the ability to inhibit thought and action: A users' guide to the stop signal paradigm.* Academic Press, 1994a. (cit. on pp. 22)

Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. (cit. on pp. 22)

Gordon D Logan. Spatial attention and the apprehension of spatial relations. *Journal of Experimental Psychology: Human Perception and Performance*, 20(5): 1015, 1994b. (cit. on pp. 22, 46)

Cathleen M. Moore, Catherine L. Elsinger, and Alejandro Lleras. Visual attention and the apprehension of spatial relations: The case of depth. *J. Exp. Psychol. Hum. Percept. Perform.*, 20(5):1015–1036, 1994. ISSN 0096-1523. (cit. on pp. 22)

Luke J Rosielle, Brian T Crabb, and Eric E Cooper. Attentional coding of categorical relations in scene perception: evidence from the flicker paradigm. *Psychon. Bull. Rev.*, 9(2):319–26, 2002. ISSN 1069-9384. (cit. on pp. 22)

Alex O. Holcombe, Daniel Linares, and Maryam Vaziri-Pashkam. Perceiving spatial relations via attentional tracking and shifting. *Curr. Biol.*, 21(13): 1135–1139, 2011. ISSN 09609822. (cit. on pp. 22)

Ineke J M Van Der Ham, Maarten J A Duijndam, Mathijs Raemaekers, Richard J A van Wezel, Anna Oleksiak, and Albert Postma. Retinotopic mapping of categorical and coordinate spatial relation processing in early visual cortex. *PLoS One*, 7(6):1–8, 2012. ISSN 19326203. (cit. on pp. 22)

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. 22)

Maria Golde, D. Yves von Cramon, and Ricarda I. Schubotz. Differential role of anterior prefrontal and premotor cortex in the processing of relational information. *Neuroimage*, 49(3):2890–2900, 2010. ISSN 10538119. (cit. on pp. 22)

Pamela E Clevenger and John E Hummel. Working memory for relations among objects. *Attention, Perception, & Psychophysics*, 76(7):1933–1953, 2014. (cit. on pp. 22)

Timothy F. Brady and George A. Alvarez. Contextual effects in visual working memory reveal hierarchically structured memory representations. *Journal of Vision*, 15(15):6, November 2015. doi: 10.1167/15.15.6. URL https://doi.org/10.1167/15.15.6. (cit. on pp. 22)

Matthew Ricci, Rémi Cadène, and Thomas Serre. Same-different conceptualization: a machine vision perspective. *Current Opinion in Behavioral Sciences*, 37:47 – 55, 2021. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2020.08.008. (cit. on pp. 22, 50, 52, 74)

Kevin Ellis, Armando Solar-Lezama, and J. Tenenbaum. Unsupervised learning by program synthesis. In *NIPS*, 2015a. (cit. on pp. 22)

Junkyung Kim, Matthew Ricci, and Thomas Serre. Not-so-clevr: learning same–different relations strains feedforward neural networks. *Interface focus*, 8(4): 20180011, 2018. (cit. on pp. 22, 23, 25, 27, 29, 30, 31, 34, 46, 50, 52, 63, 74, 75)

Nicola Messina, Giuseppe Amato, Fabio Carrara, Claudio Gennaro, and Fabrizio Falchi. Solving the same-different task with convolutional neural networks. *Pattern Recognition Letters*, 2021a. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2020.12.019. (cit. on pp. 22)

Sebastian Stabinger, David Peer, Justus Piater, and Antonio Rodríguez-Sánchez. Evaluating the progress of deep learning for visual relational concepts. *Journal of Vision*, 21(11):8–8, 10 2021. ISSN 1534-7362. doi: 10.1167/jov.21.11.8. URL https://doi.org/10.1167/jov.21.11.8. (cit. on pp. 22, 63, 75)

Sebastian Stabinger, Antonio Rodríguez-Sánchez, and Justus Piater. 25 years of cnns: Can we compare to human abstraction capabilities? In Alessandro E.P. Villa, Paolo Masulli, and Antonio Javier Pons Rivero, editors, *Artificial Neural Networks and Machine Learning – ICANN 2016*, pages 380–387, Cham, 2016a. Springer International Publishing. ISBN 978-3-319-44781-0. (cit. on pp. 22, 23, 25, 75)

Guillermo Puebla and Jeffrey S. Bowers. Can deep convolutional neural networks learn same-different relations? *bioRxiv*, 2021. doi: 10.1101/2021.04.06.438551. URL https://www.biorxiv.org/content/early/2021/05/12/2021.04.06.438551. (cit. on pp. 22, 23, 63, 74, 75)

Christina M. Funke, Judy Borowski, Karolina Stosio, Wieland Brendel, Thomas S. A. Wallis, and Matthias Bethge. Five points to check when comparing visual perception in humans and machines. *Journal of Vision*, 21(3):16, March 2021a. doi: 10.1167/jov.21.3.16. URL https://doi.org/10.1167/jov.21.3.16. (cit. on pp. 22)

Lu Yihe, Scott C Lowe, Penelope A Lewis, and Mark CW van Rossum. Program synthesis performance constrained by non-linear spatial relations in synthetic visual reasoning test. *arXiv preprint arXiv:1911.07721*, 2019a. (cit. on pp. 23, 25, 30, 75)

Andrea Alamia, Canhuang Luo, Matthew Ricci, Junkyung Kim, Thomas Serre, and Rufin VanRullen. Differential involvement of eeg oscillatory components in sameness versus spatial-relation visual reasoning tasks. *eNeuro*, 8 (1), 2021a. doi: 10.1523/ENEURO.0267-20.2020. URL https://www.eneuro.org/content/8/1/ENEURO.0267-20.2020. (cit. on pp. 23, 31, 46, 47)

Chaz Firestone. Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, 117(43):26562–26571, 2020. (cit. on pp. 24)

Kenneth D Forbus and Andrew Lovett. Same/different in visual reasoning. *Current Opinion in Behavioral Sciences*, 37:63–68, 2021. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2020.09.008. URL https://www.sciencedirect.com/science/article/pii/S2352154620301431. Same-different conceptualization. (cit. on pp. 24)

Dedre Gentner, Ruxue Shao, Nina Simms, and Susan Hespos. Learning same and different relations: cross-species comparisons. *Current Opinion in Behavioral Sciences*, 37:84–89, 2021a. ISSN 2352-1546. doi: https://doi.org/10.1016/j.cobeha.2020.11.013. URL https://www.sciencedirect.com/science/article/pii/S2352154620301728. Same-different conceptualization. (cit. on pp. 24)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (cit. on pp. 25, 58, 60)

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learn-

ing in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. (cit. on pp. 28)

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021. (cit. on pp. 30)

Nicola Messina, Giuseppe Amato, Fabio Carrara, Claudio Gennaro, and Fabrizio Falchi. Recurrent vision transformer for solving visual reasoning problems, 2021b. (cit. on pp. 30, 50, 52, 60, 63, 75, 77)

Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. *arXiv preprint arXiv:1806.10909*, 2018. (cit. on pp. 31)

Gary F. Marcus. *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. The MIT Press, 04 2001. ISBN 9780262279086. doi: 10.7551/mitpress/1187.001.0001. URL https://doi.org/10.7551/mitpress/1187.001.0001. (cit. on pp. 31)

Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*, 2018a. (cit. on pp. 31)

Drew Linsley, Junkyung Kim, Alekh Ashok, and Thomas Serre. Recurrent neural circuits for contour detection. *arXiv preprint arXiv:2010.15314*, 2020. (cit. on pp. 31)

Kimberly Villalobos, Vilim Štih, Amineh Ahmadinejad, Shobhita Sundaram, Jamell Dozier, Andrew Francl, Frederico Azevedo, Tomotake Sasaki, and Xavier Boix. Do neural networks for segmentation understand insideness? *Neural Computation*, 33(9):2511–2549, 2021. (cit. on pp. 31)

Robert Egly, Robert Rafal, Jon Driver, and Yves Starrveveld. Covert orienting in the split brain reveals hemispheric specialization for object-based attention. *Psychological science*, 5(6):380–383, 1994b. (cit. on pp. 34, 35, 47)

Pieter R Roelfsema, Victor AF Lamme, and Henk Spekreijse. Object-based attention in the primary visual cortex of the macaque monkey. *Nature*, 395(6700): 376–381, 1998. (cit. on pp. 34, 35)

Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015. (cit. on pp. 35)

Kan Chen, Jiang Wang, Liang - Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. ABC-CNN: an attention based convolutional neural network for visual question answering. *CoRR*, abs / 1511.05960, 2015. URL http://arxiv.org/abs/1511.05960. (cit. on pp. 35)

Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29, 2016. (cit. on pp. 35)

Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *CoRR*, abs / 1511.05234, 2015. URL http://arxiv.org/abs/1511.05234. (cit. on pp. 35)

Mengye Ren and Richard S. Zemel. End-to-end instance segmentation and counting with recurrent attention. *CoRR*, abs / 1605.09410, 2016. URL http://arxiv.org/abs/1605.09410. (cit. on pp. 35)

Marijn F Stollenga, Jonathan Masci, Faustino Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. In *Advances in neural information processing systems*, pages 3545–3553, 2014. (cit. on pp. 35)

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017. (cit. on pp. 35)

Drew Linsley, Dan Scheibler, Sven Eberhardt, and Thomas Serre. Global-and-local attention networks for visual recognition. *arXiv preprint arXiv:1805.08819*, 2018b. (cit. on pp. 35)

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. (cit. on pp. 35)

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for End-to-End object detection. October 2020. (cit. on pp. 35)

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. Attention over learned object embeddings enables complex visual reasoning. volume 34, pages 9112–9124, 2021. (cit. on pp. 35)

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. (cit. on pp. 36)

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. (cit. on pp. 46)

John K. Tsotsos, Antonio Jose Rodriguez-Sanchez, Albert L. Rothenstein, and Eugene Simine. Different binding strategies for the different stages of visual recognition. In Francesco Mele, Giuliana Ramella, Silvia Santillo, and Francesco Ventriglia, editors, *Advances in Brain, Vision, and Artificial Intelligence*, pages 150–160, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-75555-5. (cit. on pp. 46)

Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45, 1997. (cit. on pp. 50)

Andrew Lovett and Kenneth Forbus. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1):60, 2017. (cit. on pp. 50)

Andrea Alamia, Canhuang Luo, Matthew Ricci, Junkyung Kim, Thomas Serre, and Rufin VanRullen. Differential involvement of eeg oscillatory components in sameness versus spatial-relation visual reasoning tasks. *eNeuro*, 8 (1), 2021b. (cit. on pp. 50, 51, 75)

Shimon Ullman. Visual routines. *Cognition*, 18:97–159, 1984. (cit. on pp. 51)

Shimon Ullman. Visual routines. In *Readings in computer vision*, pages 298–328. Elsevier, 1987. (cit. on pp. 51)

Mary Hayhoe. Vision using routines: A functional account of vision. *Visual Cognition*, 7(1-3):43–64, 2000. (cit. on pp. 51)

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. 51, 75)

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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. (cit. on pp. 51, 75)

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Hierarchical rule induction network for abstract visual reasoning. *arXiv preprint arXiv:2002.06838*, 2(4), 2020. (cit. on pp. 51)

Steven Spratley, Krista Ehinger, and Tim Miller. A closer look at generalisation in raven. In *European Conference on Computer Vision*, pages 601–616. Springer, 2020. (cit. on pp. 51)

Sebastian Stabinger, Antonio Rodríguez-Sánchez, and Justus Piater. 25 years of cnns: Can we compare to human abstraction capabilities? In *International Conference on Artificial Neural Networks*, pages 380–387. Springer, 2016b. (cit. on pp. 52, 63, 75)

Lu Yihe, Scott C. Lowe, Penelope A. Lewis, and Mark C. W. van Rossum. Program synthesis performance constrained by non-linear spatial relations in synthetic visual reasoning test. *ArXiv*, abs/1911.07721, 2019b. (cit. on pp. 52)

Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017. (cit. on pp. 52, 53, 57, 67, 74)

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers and distillation through attention. In *International Conference on Machine Learning*, volume 139, pages 10347–10357, July 2021b. (cit. on pp. 52)

Taylor Webb, Zachary Dulberg, Steven Frankland, Alexander Petrov, Randall O'Reilly, and Jonathan Cohen. Learning representations that support extrap-

olation. In *International Conference on Machine Learning*, pages 10136–10146. PMLR, 2020. (cit. on pp. 53)

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6281–6290, 2019. (cit. on pp. 55)

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. (cit. on pp. 58, 60)

C. M. Funke, J. Borowski, K. Stosio, W. Brendel, T. S. A. Wallis, and M. Bethge. Five points to check when comparing visual perception in humans and machines. *Journal of Vision*, Mar 2021b. URL https://jov.arvojournals.org/article.aspx?articleid=2772393. (cit. on pp. 60)

Nicola Messina, Giuseppe Amato, Fabio Carrara, Claudio Gennaro, and Fabrizio Falchi. Solving the same-different task with convolutional neural networks. *Pattern Recognition Letters*, 143:75–80, 2021c. (cit. on pp. 60, 74)

Lukas Hahne, Timo Lüddecke, Florentin Wörgötter, and David Kappel. Attention on abstract visual reasoning. *arXiv preprint arXiv:1911.05990*, 2019. (cit. on pp. 62)

Kevin Ellis, Armando Solar-Lezama, and Joshua B. Tenenbaum. Unsupervised learning by program synthesis. In *NIPS*, 2015b. (cit. on pp. 63, 75)

Dedre Gentner, Ruxue Shao, Nina Simms, and Susan Hespos. Learning same and different relations: cross-species comparisons. *Current Opinion in Behavioral Sciences*, 37:84–89, 2021b. (cit. on pp. 74)

Martin Giurfa, Shaowu Zhang, Arnim Jenett, Randolf Menzel, and Mandyam V Srinivasan. The concepts of 'sameness' and 'difference'in an insect. *Nature*, 410(6831):930–933, 2001. (cit. on pp. 74)

Antone Martinho and Alex Kacelnik. Ducklings imprint on the relational concept of "same or different". *Science*, 353(6296):286–288, 2016. (cit. on pp. 74)

Aimen Zerroug, Mohit Vaishnav, Julien Colin, Sebastian Musslick, and Thomas Serre. A benchmark for compositional visual reasoning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2022. (cit. on pp. 75, 77)

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. (cit. on pp. 75)

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019. (cit. on pp. 75)

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021. (cit. on pp. 77)