UNIVERSITY OF CALIFORNIA

Los Angeles

Few-shot Concept Induction
through Lenses of Intelligence Quotient Tests

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Chi Zhang

2022

ABSTRACT OF THE DISSERTATION

Few-shot Concept Induction
through Lenses of Intelligence Quotient Tests

by

Chi Zhang
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2022
Professor Song-Chun Zhu, Chair

Humans not only *learn* concepts from labeled supervision but also *induce* new relational concepts unsupervisedly from observing reoccurring sequences of events. In contrast with the abundance of tasks that challenge machines on perception, one that evaluates machines' few-shot concept induction ability has been long overdue. To endow machines with such capability and fill the missing gap, we start with the introduction of RAVEN, a dataset based on the cognitive study of Raven's Progressive Matrices (RPM) that has proven to be effective in measuring humans' few-shot concept induction. In particular, we note that neural methods that are supplied with the idea of contrastive learning can significantly improve both model performance and learning efficiency. However, completely neural methods are neither interpretable nor performative. Therefore, we further propose neuro-symbolic approaches. We first introduce a neuro-symbolic Probabilistic Abduction and Execution (PrAE) learner; central to the PrAE learner is the process of probabilistic abduction and execution on a probabilistic scene representation, akin to the mental manipulation of objects. In PrAE, we disentangle perception and reasoning from a monolithic model. The neural visual perception frontend predicts objects' attributes, later aggregated by a scene inference engine to produce a probabilistic scene representation. In the symbolic logical reasoning backend, the PrAE learner uses the representation to abduce the hidden rules. An answer is predicted by executing the rules on the probabilistic representation. The entire system is trained end-to-end in an analysis-by-synthesis manner without any visual attribute annotations. While effective, PrAE essentially turns the induction problem into abduction problem as explicit knowledge is recruited. We then introduce the ALgebra-Aware Neuro-Semi-Symbolic (ALANS) learner. The ALANS learner is motivated by abstract algebra and the representation theory. It consists of a neural visual perception frontend and an algebraic abstract reasoning backend: the frontend summarizes the visual information from object-based representation, while the backend transforms it into an algebraic structure and induces the hidden operator on the fly. The induced operator is later executed to predict the answer's representation, and the choice most similar to the prediction is selected as the solution. Both methods explicitly realize the computational process of reasoning, achieve improved performance, and are more interpretable and easy for debugging. However, compared to PrAE, ALANS fully implements the induction process as on-the-fly optimization. Experiments show that the ALANS learner outperforms various pure connectionist models in domains requiring systematic generalization. We further show the generative nature of the learned algebraic representation; it can be decoded by isomorphism to generate an answer. The results and analysis demonstrate that the learned algebraic architecture facilitates relational learning and is a viable schema for few-shot concept learning.

The dissertation of Chi Zhang is approved.

Quanquan Gu

Hongjing Lu

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2022

To my mother and father

who provided unconditonal support

along the arduous journey

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I feel especially lucky to have most of my work accepted almost immediately after the initial submission, and others timely received before graduation. For this aspect of my Ph.D., I would like to thank the following people:

Prof. Song-Chun Zhu, my advisor, for continually supporting and funding my research. Prof. Zhu is never an easy-going person. He is tough but his visionary should not be overshadowed by his temper. Instead, I'm always impressed by his taste of research, his ambition, and his never-ending pursuit in his goal.

Prof. Demetri Terzopoulos, for introducing me to the field of computer graphics and generously serving as a committee member. I've always admired his contribution to the Academy and I feel regretful not to have extensive collaboration with him on additional projects.

Prof. Hongjing Lu, for those insightful and fruitful discussions we had pre-COVID. Prof. Lu opened the door of Psychology for me and for a student with the engineering background, you couldn't imagine how amazing it was to learn the designs of serious Psychology studies.

Prof. Quanquan Gu, for being a role model in conducting rigorous theoretic research. It was a no-discord-no-concord experience, and through that I realize Prof. Gu is really strict with students. I believe that is the reason Prof. Gu and his team produce a series of high-impact research.

Prof. Ying Nian Wu, the nicest and kindest person I've ever met. From Prof. Wu, I realize the beauty of research and how one can really enjoy his work. He is the most patient teacher, and if anything, he should also be called an educator.

Prof. Yixin Zhu, for being a long-term friend and collaborator outside and inside our lab. We first met the summer I visited VCLA and later became research buddies for the entirety of my Ph.D. Yixin has always shielded me from troubles along the five-year journey, and served as my friend, my collaborator, my mentor, and my manager at VCLA, CARA, and DMAI. I hope our friendship continues at PKU.

Dr. Feng Gao, for being a reliable and trustable friend, collaborator, and roommate. Like Yixin, Feng is another one that plays an important role during my Ph.D., not only for his collaboration in research, but also for helping me settle down in Los Angeles.

Dr. Mark Edmunds, for introducing me to the American culture and ideology. I really miss the days when Mark, Feng, and I heatedly discussed the political and societal issues. While most of the times we viewed the same problem from drastically different angles, I did find his opinions valuable. And that introduced me to a whole new perspective.

Dr. Xu Xie, for offering me free rides during my Ph.D. and bringing his cute dog to help me with my depression. Xu also offered me a lot of career options and advice when I was looking for opportunities. Though I didn't landed in any, I would still like to thank for his help.

Dr. Hangxin Liu, for helping me through my early days at VCLA. As an officemate, Hangxin is the first collaborator of mine at VCLA and continues to be a mentor at DMAI. He is a significant factor in my career decision and I hope to have continual collaboration.

Mr. Sirui Xie, for being a fun roommate, a sharp-minded collaborator, and a nice friend. Sirui is the guy I spent most time with during COVID and he was the biggest emotional support, the one to talk to and have fun with, at the quarantine time. I enjoyed playing Overcooked with Sirui.

Mr. Baoxiong Jia, for bringing me so much fun and joy when working together. I don't know where his jokes come from, but every time we talked, he would come up with some nice funny jokes and I couldn't help laughing.

Ms. Yining Hong, Ms. Yuxin Qiu, Ms. Shuwen Qiu. They are the few girls in our lab. Yining has been so productive and I really admire her power. I lived with Yuxin and Feng for quite a while during COVID. They are a cute couple, and I would also like to thank for sharing their cat, Mianhuan, with me. Shuwen is really considerate and cooks really well. I still remember her sharing good food with me at my down time.

Dr. Zilong Zeng, Dr. Siyuan Huang, Dr. Yixin Chen, Dr. Tengyu Liu, Dr. Qing Li, Dr Lifeng Fan for providing me career support when I was about to graduate.

Mr. Ziyuan Jiao, Mr. Zeyu Zhang, Mr. Muzhi Han, for introducing the field of robotics to me.

Mr. Xiaojian Ma, for exchanging those practical ideas.

Dr. Baoyuan Liu, for being a nice and helpful mentor when I interned at Amazon.

Mr. Frederic Liu, neither a direct manager nor a teammate. But without his efforts, I couldn't have done my job at Google, and I would also like to thank for his generous help during my Google conversion.

Looking back to the year when I decided to pursue a Ph.D. degree, I couldn't help getting emotional. 2017 was still a year of globalization and 2022 has shown serious signs of polarization plus pandemic. For an individual, the ever-changing landscape has captured me feeling lost and confused on my future path. However, it is especially at this time when I received the most generous peer support over the years.

Yixin, Feng, Mark, and Xu are the ones who introduced me to Los Angeles when I first arrived. Before pandemic and when I didn't have a car, they carried me around and we together explored the various aspects of Los Angeles. They took me to authentic restaurants when I missed my hometown. Feng even taught me how to pass the driving test when I failed the first road test.

The pandemic has brought us closer. Shuwen, Sirui, Baoxiong, Ziyuan, and Zeyu all moved in to the university apartment and we started to have regular get-together. Feng, Yuxin, and I usually had hot pot at home once a week when we shared food bought from the 99 Ranch. Xu occasionally walked his dog in the yard and when he did so, we would all stepped out of the room and played with the energetic golden retriever. Mark had been constantly traveling across the country but when he dropped by, we would have authentic

Chinese food that he liked. At the beginning of the pandemic, I believed I developed signs of depression due to lack of interaction with people. Baoxiong, Sirui, Feng, Yuxin, and I started to have virtual Friday night party, and Feng and Yuxin had a cat named Mianhua that I could have fun with. Luckily, I recovered. Were it not for the help from my friends, the symptoms might have become worse.

As the pandemic eased, Sirui moved in to my apartment and we regularly visited supermarkets together. Feng, Yuxin, Qing, Shuwen, Sirui and I later even had road trips together. The pandemic also saw our cooking skills skyrocketing (honestly we were planning some cooking contest when we returned to school, which unfortunately may not take place with the entire group). Later, Lifeng introduced us to a new friend Yuxi Ma, who magically became a significant friend of my during Ph.D. and very likely still so when I onboard my new job.

I still remembered the night when I was about to leave Los Angeles for my internship aboard. All the labmates in the apartment joined and everyone made their specialty dishes. They saw me off at LAX, and I was under the impression that we would just meet in one year.

Unfortunately, we might not meet in a short time. When I boarded the plane, I was happy to be able to go home and see my family three years since I visited them. But unexpectedly, that might be the last time for my friends in the coming years.

I feel sorry now not to properly say goodbye at that time. But whether or not I can see them in a short time, I would like to say thanks for every one in the lab for creating the most wonderful five years for my Ph.D.

VITA

2016	Visiting scholar at VCLA with Prof. Song-Chun Zhu.
2013–2017	B.E. (Computer Science), Zhejiang University.
2017-2019	M.S. (Computer Science), UCLA.
2017-2019	Graduate Student Researcher at VCLA with Prof. Song-Chun Zhu.
2019	Ph.D. candidate in Computer Science, UCLA.
2020-2021	Teaching Assistant, Computer Science, UCLA.
2021	Applied Scientist Intern, Amazon.
2021	Research Intern, Google
2021–present	Research Scientist Intern, Beijing Institute for General Artificial Intelligence (BIGAI).

PUBLICATIONS

Mirroring without overimitation: Learning functionally equivalent manipulation actions. Liu, H., **Zhang, C.**, Zhu, Y., Jiang, C., Zhu, S. C. *AAAI*, 2019.

Metastyle: Three-way trade-off among speed, flexibility, and quality in neural style transfer. **Zhang, C.**, Zhu, Y., Zhu, S. C. *AAAI*, 2019.

Raven: A dataset for relational and analogical visual reasoning. **Zhang, C.**, Gao, F., Jia, B., Zhu, Y., Zhu, S. C. *CVPR*, 2019.

Learning virtual grasp with failed demonstrations via bayesian inverse reinforcement learning. Xie, X., Li, C., **Zhang, C.**, Zhu, Y., Zhu, S. C. *IROS*, 2019.

Learning perceptual inference by contrasting. **Zhang, C.**, Jia, B., Gao, F., Zhu, Y., Lu, H., Zhu, S. C. *NeurIPS*, 2019.

Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning. Zhang, W., **Zhang, C.**, Zhu, Y., Zhu, S. C. *AAAI*, 2020.

Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. Zhu, Y., Gao, T., Fan, L., Huang, S., Edmonds, M., Liu, H., Gao, F., **Zhang, C.**, Qi, S., Wu, Y. N., Zhu, S. C. *Enginnering 6*, 2020.

Congestion-aware multi-agent trajectory prediction for collision avoidance. Xie, X., **Zhang, C.**, Zhu, Y., Wu, Y. N., Zhu, S. C. *ICRA*, 2021.

Abstract spatial-temporal reasoning via probabilistic abduction and execution. **Zhang, C.**, Jia, B., Zhu, S. C., Zhu, Y. *CVPR*, 2021.

Acre: Abstract causal reasoning beyond covariation. **Zhang, C.**, Jia, B., Edmonds, M., Zhu, S. C., Zhu, Y. *CVPR*, 2021.

Learning algebraic representation for systematic generalization in abstract reasoning. **Zhang, C.**, Xie, S., Jia, B., Wu, Y. N., Zhu, S. C., Zhu, Y. *ECCV*, 2022.

CHAPTER 1

Introduction

Tremendous success has been witnessed over the years in artificial intelligence and machine learning on perceptual tasks with considerable amount of well-prepared data: with sufficient input-output labels, machine systems have managed to classify images, detect objects, generate realistic photos from text, answer questions, and play games; unsupervised methods have even produced language models that can fluently generate reasonable human utterances reflecting commonsense and serve as "foundation" for general-purpose learning. However, the ability of relational induction, paramount to human core intelligence, has paradoxically escaped under our nose. Apart from learning new concepts from supervision of input-output label pairs, more crucially, we can induce new concepts from regularly occurring events. While the former can be captured using function approximators like neural networks, how to realize human-level few-shot concept induction remains a challenge for the community to solve.

In this work, we detail a series of exploration to build machines that are capable of few-shot concept induction.

The first challenge we met is the lack of a comprehensive evaluation benchmark for few-shot concept induction. To fill in this gap, we first introduce RAVEN, a synthetic diagnostic dataset for few-shot concept induction built on a popular task in IQ tests called Raven's Progressive Matrices (RPM). In this task, a test taker needs to first induce the hidden relations from the limited number of context panels and use the induced relation to find the correct answer in the missing panel. Human performance on this task is believed to be correlated with the general intelligence. Preliminary experiments show that existing data-driven methods are still inferior in the task.

As a first step towards the goal, we follow the data-driven paradigm and recruit the idea of contrastive learning to propose Contrastive Perceptual Inference Network (CoPINet). Contrastive learning is baked into CoPINet in two levels: the module level and the loss level. In particular, we introduce the constrast module and the contrast loss. The contrast module first summarizes features from different choices and then subtracts the common features from each choice. The constrast loss leverages the idea in Noise-Contrastive Estimation (NCE) and encourages the information gap between the correct choice and a wrong choice to be infinitely large. Experimental results show that constrastive learning improves both the final performance and learning efficiency.

However, the CoPINet architecture does not perform any explicit form of reasoning. We therefore introduce a neuro-symbolic method called Probabilistic Abduction and Execution (PrAE) that explicitly models the perception process and the reasoning process. In the perception process, PrAE uses a CNN-based neural network to detect object attributes. The attributes are kept as probabilistic states and fed into the following reasoning step. The reasoning engine retrieves additional knowledge to perform planning-like differentiable one-step reasoning. It first performs inverse planning to find the hidden relations and then uses the relations to predict the answer's probabilistic representation. The choice most similar is selected as the answer. We find that in experiments meaningful visual representation emerges during joint training and we can probe into both its perception performance and reasoning performance.

We finally introduce teh ALgebra-Aware Neuro-Semi-Symbolic (ALANS) learner, a neuro-semi-symbolic architecture to perform few-shot concept induction. ALANS follows the general neuro-symbolic framework of PrAE but relaxes the constraint on outside knowledge and implements induction as on-the-fly optimization. The model is built on the algebraic representation from the Peano Axiom and the general representation theory. In experiments, we find that ALANS not only achieves the best performance in the traditional Independent and Identically Distributed (I.I.D.) setup where the training set and the test set follow the same distribution, but also outperform all other baselines on the systematic generalization splits. Besides, apart from enjoying PrAE's benefits, ALANS can also be turned into a generative model that can directly predict the answer representation and use an off-the-shelf rendering engine to generate the answer.

The results and analysis demonstrate that the ALANS architecture facilitates relational learning and is a viable schema for few-shot concept learning.

CHAPTER 2

RAVEN: A Dataset for Measuring Few-shot Concept Induction

Dramatic progress has been witnessed in basic vision tasks involving low-level perception, such as object recognition, detection, and tracking. Unfortunately, there is still an enormous performance gap between artificial vision systems and human intelligence in terms of higher-level vision problems, especially ones involving reasoning. Earlier attempts in equipping machines with high-level reasoning have hovered around Visual Question Answering (VQA), one typical task associating vision and language understanding. In this work, we propose a new dataset, built in the context of RPM and aimed at lifting machine intelligence by associating vision with structural, relational, and analogical reasoning in a hierarchical representation. Unlike previous works in measuring abstract reasoning using RPM, we establish a semantic link between vision and reasoning by providing structure representation. This addition enables a new type of abstract reasoning by jointly operating on the structure representation. Machine reasoning ability using modern computer vision is evaluated in this newly proposed dataset. Additionally, we also provide human performance as a reference. Finally, we show consistent improvement across all models by incorporating a simple neural module that combines visual understanding and structure reasoning.

2.1 Introduction

> The study of vision must therefore include not only the study of how to extract from images . . . , but also an inquiry into the nature of the *internal representations* by which we *capture* this information and thus make it available as a **basis** for *decisions about our thoughts and actions.*
>
> — David Marr, 1982 [Mar82]

Computer vision has a wide spectrum of tasks. Some computer vision problems are clearly purely visual, "capturing" the visual information process; for instance, filters in early vision [CR68], primal sketch [GZW07] as the intermediate representation, and Gestalt laws [KK79] as the perceptual organization. In contrast, some other vision problems have trivialized requirements for perceiving the image, but engage more generalized problem-solving in terms of relational and/or analogical visual reasoning [HHT96]. In such cases, the vision component becomes the "basis for decisions about our thoughts and actions".

Currently, the majority of the computer vision tasks focus on "capturing" the visual information process; few lines of work focus on the later part—the relational and/or analogical visual reasoning. One existing line of work in equipping artificial systems with reasoning ability hovers around Visual Question Answering (VQA) [AAL15, JHM17a, RKZ15, YWG18, ZGB16]. However, the reasoning skills required in VQA lie only at the periphery of the cognitive ability test circle [CJS90]. To push the limit of computer vision or more broadly speaking, Artificial Intelligence (AI), towards the center of cognitive ability test circle, we need a test originally designed for measuring human's intelligence to challenge, debug, and improve the current artificial systems.

A surprisingly effective ability test of human visual reasoning has been developed and identified as the Raven's Progressive Matrices (RPM) [KMG13, Rav38, SCS13], which is widely accepted and believed to be highly correlated with real intelligence [CJS90]. Unlike VQA, RPM lies directly at the center of human intelligence [CJS90], is diagnostic of abstract

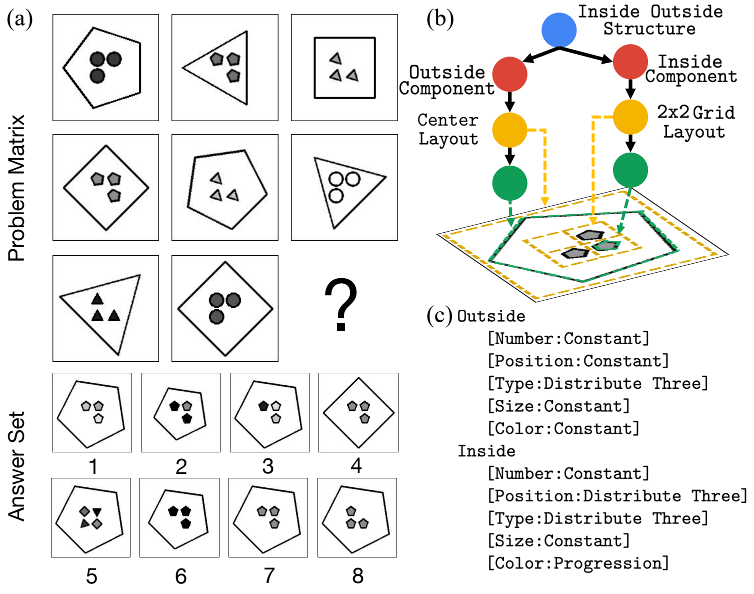


Figure 2.1: (a) An example RPM. One is asked to select an image that best completes the problem matrix, following the structural and analogical relations. Each image has an underlying structure. (b) Specifically in this problem, it is an inside-outside **structure** in which the outside **component** is a layout with a single centered object and the inside **component** is a 2 × 2 grid layout. Details in Fig. 2.2. (c) lists the rules for (a). The compositional nature of the rules makes this problem a difficult one. The correct answer is 7.

and structural reasoning ability [EKM84], and characterizes the defining feature of high-level intelligence, *i.e.*, *fluid intelligence* [JBJ08].

Fig. 2.1 shows an example of RPM problem together with its structure representation. Provided two rows of figures consisting of visually simple elements, one must efficiently derive the correct image structure (Fig. 2.1(b)) and the underlying rules (Fig. 2.1(c)) to jointly reason about a candidate image that best completes the problem matrix. In terms of levels of reasoning required, RPM is arguably harder compared to VQA:

- Unlike VQA where natural language questions usually imply what to pay attention to in the image, RPM relies merely on visual clues provided in the matrix and the *correspondence* *problem* itself, *i.e.*, finding the correct level of attributes to encode, is already a major factor distinguishing populations of different intelligence [CJS90].

- While VQA only requires spatial and semantic understanding, RPM needs joint spatial-temporal reasoning in the problem matrix and the answer set. The limit of *short-term memory*, the ability of *analogy*, and the discovery of the *structure* have to be taken into consideration.
- Structures in RPM make the compositions of rules much more complicated. Unlike VQA whose questions only encode relatively simple first-order reasoning, RPM usually includes more sophisticated logic, even with recursions. By composing different rules at various levels, the reasoning progress can be extremely difficult.

To push the limit of current vision systems' reasoning ability, we generate a new dataset to promote further research in this area. We refer to this dataset as the Relational and Analogical Visual rEasoNing dataset (RAVEN) in homage to John Raven for the pioneering work in the creation of the original RPM [Rav38]. In summary:

- RAVEN consists of $1,120,000$ images and $70,000$ RPM problems, equally distributed in 7 distinct figure configurations.
- Each problem has 16 tree-structure annotations, totaling up to $1,120,000$ structural labels in the entire dataset.
- We design 5 rule-governing attributes and 2 noise attributes. Each rule-governing attribute goes over one of 4 rules, and objects in the same component share the same set of rules, making in total $440,000$ rule annotations and an average of 6.29 rules per problem.

The RAVEN dataset is designed inherently to be light in visual recognition and heavy in reasoning. Each image only contains a limited set of simple gray-scale objects with clear-cut boundaries and no occlusion. In the meantime, rules are applied row-wise, and there could be one rule for each attribute, attacking visual systems' major weaknesses in *short-term memory* and *compositional reasoning* [JHM17a].

An obvious paradox is: in this innately compositional and structured RPM problem, no annotations of structures are available in previous works (*e.g.*, [BHS18, WS15]). Hence, we set out to establish a semantic link between visual reasoning and structure reasoning in RPM. We ground each problem instance to a sentence derived from an Attributed Stochastic Image Grammar (A-SIG) [Fu74, LWP09, PZ15, WXZ07, ZWZ16, ZM07] and decompose the data generation process into two stages: the first stage samples a sentence from a pre-defined A-SIG and the second stage renders an image based on the sentence. This structured design makes the dataset very diverse and easily extendable, enabling generalization tests in different figure configurations. More importantly, the data generation pipeline naturally provides us with abundant dense annotations, especially the structure in the image space. This semantic link between vision and structure representation opens new possibilities by breaking down the problem into image understanding and tree- or graph-level reasoning [KW16, TSM15]. As shown in Sec. 2.6, we empirically demonstrate that models with a simple structure reasoning module to incorporate both vision-level understanding and structure-level reasoning would notably improve their performance in RPM.

The organization of the paper is as follows. In Sec. 2.2, we discuss related work in visual reasoning and computational efforts in RPM. Sec. 2.3 is devoted to a detailed description of the RAVEN dataset generation process, with Sec. 2.4 benchmarking human performance and comparing RAVEN with a previous RPM dataset. In Sec. 2.5, we propose a simple extension to existing models that incorporates vision understanding and structure reasoning. All baseline models and the proposed extensions are evaluated in Sec. 2.6. The notable gap between human subjects (84%) and vision systems (59%) calls for further research into this problem. We hope RAVEN could contribute to the long-standing effort in human-level reasoning AI.

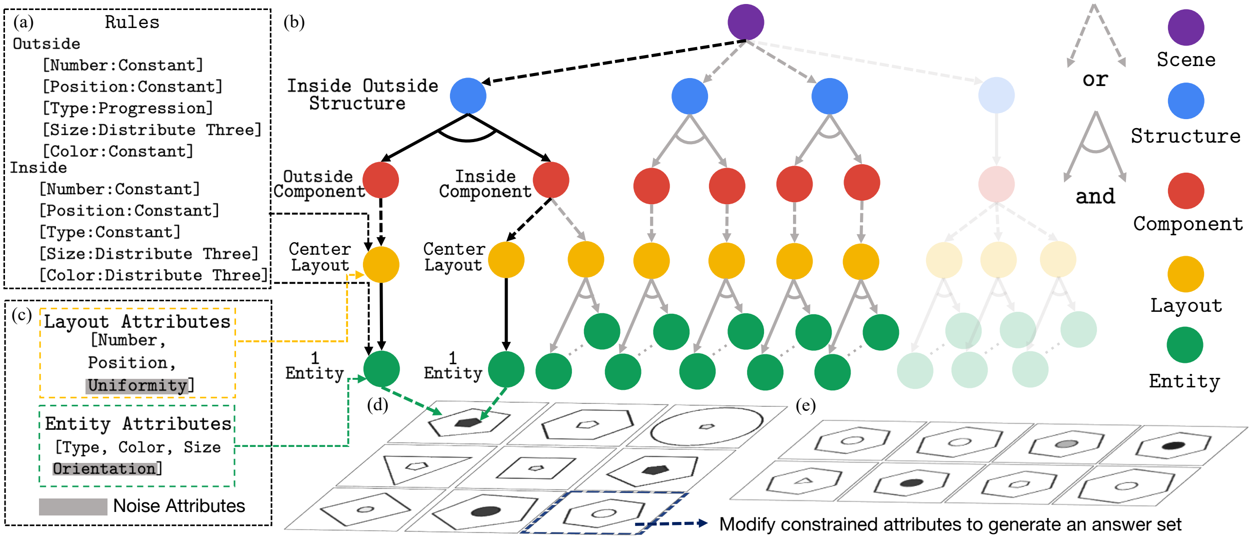


Figure 2.2: RAVEN creation process. A graphical illustration of the grammar production rules used in A-SIG is shown in (b). Note that `Layout` and `Entity` have associated attributes (c). Given a randomly sampled rule combination (a), we first prune the grammar tree (the transparent branch is pruned). We then sample an image structure together with the values of the attributes from (b), denoted by black, and apply the rule set (a) to generate a single row. Repeating the process three times yields the entire problem matrix in (d). (e) Finally, we sample constrained attributes and vary them in the correct answer to break the rules and obtain the candidate answer set.

2.2 Related Work

Visual Reasoning Early attempts were made in 1940s-1970s in the field of logic-based AI. Newell argued that one of the potential solutions to AI was "to construct a single program that would take a standard intelligence test" [New73]. There are two important trials: (i) Evans presented an AI algorithm that solved a type of geometric analogy tasks in the Wechsler Adult Intelligence Scale (WAIS) test [Eva62, Eva64], and (ii) Simon and Kotovsky devised a program that solved Thurstone letter series completion problems [TT41]. However, these early attempts were heuristic-based with hand-crafted rules, making it difficult to apply to other problems.

The reasoning ability of modern vision systems was first systematically analyzed in the CLEVR dataset [JHM17a]. By carefully controlling inductive bias and slicing the vision systems' reasoning ability into several axes, Johnson *et al*. successfully identified major drawbacks of existing models. A subsequent work [JHM17b] on this dataset achieved good performance by introducing a program generator in a structured space and combining it with a program execution engine. A similar work that also leveraged language-guided structured reasoning was proposed in [HAR17]. Modules with special attention mechanism were latter proposed in an end-to-end manner to solve this visual reasoning task [HM18, SRB17, ZZH17]. However, superior performance gain was observed in very recent works [CLL18, MTS18, YWG18] that fell back to structured representations by using primitives, dependency trees, or logic. These works also inspire us to incorporate structure information into solving the RPM problem.

More generally, Bisk *et al*. [BSC18] studied visual reasoning in a 3D block world. Perez *et al*. [PSD18] introduced a conditional layer for visual reasoning. Aditya *et al*. [AYB18] proposed a probabilistic soft logic in an attention module to increase model interpretability. And Barrett *et al*. [BHS18] measured abstract reasoning in neural networks.

Computational Efforts in RPM The research community of cognitive science has tried to attack the problem of RPM with computational models earlier than the computer science community. However, an oversimplified assumption was usually made in the experiments that the computer programs had access to a symbolic representation of the image and the operations of rules [CJS90, LF17, LFU10, LTF09]. As reported in Sec. 2.4.4, we show that giving this critical information essentially turns it into a searching problem. Combining it with a simple heuristics provides us an optimal solver, easily surpassing human performance. Another stream of AI research [LLG12, MG14, MKG14, MSD18, SG18b] tries to solve RPM by various measurements of image similarity. To promote fair comparison between computer programs and human subjects in a data-driven manner, Wang and Su [WS15] first proposed a systematic way of automatically generating RPM using first-order logic. Barrett *et al.* [BHS18] extended their work and introduced the PGM dataset by instantiating each rule with a relation-object-attribute tuple. Hoshen and Werman [HW17] first trained a CNN to complete the rows in a simplistic evaluation environment, while Barrett *et al.* [BHS18] used an advanced Wild Relational Network (WReN) and studied its generalization.

2.3 Creating RAVEN

Our work is built on prior work aforementioned. We implement all relations in Advanced Raven's Progressive Matrices identified by Carpenter *et al.* [CJS90] and generate the answer set following *the monotonicity of RPM's constraints* proposed by Wang and Su [WS15].

Fig. 2.2 shows the major components of the generation process. Specifically, we use the A-SIG as the representation of RPM; each RPM is a parse tree that instantiates from the A-SIG. After rules are sampled, we prune the grammar to make sure the relations could be applied on any sentence sampled from it. We then sample a sentence from the pruned grammar, where rules are applied to produce a valid row. Repeating such a process three times yields a problem matrix. To generate the answer set, we modify attributes on the correct answer such that the relationships are broken. Finally, the structured presentation is fed into a rendering engine to generate images. We elaborate the details below.

2.3.1 Defining the Attributed Grammar

We adopt an A-SIG as the hierarchical and structured image grammar to represent the RPM problem. Such representation is advanced compared with prior work (*e.g.*, [BHS18, WS15]) which, at best, only maintains a flat representation of rules.

See Fig. 2.2 for a graphical illustration of the grammar production rules. Specifically, the A-SIG for RPM has 5 levels—`Scene`, `Structure`, `Component`, `Layout`, and `Entity`. Note

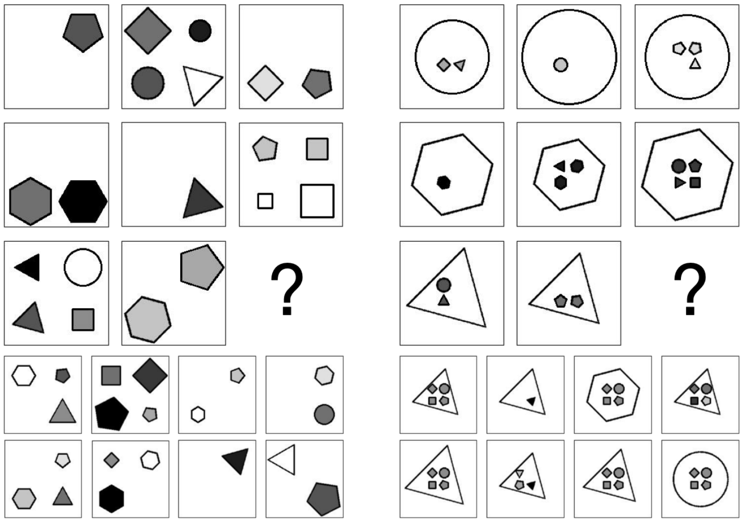


Figure 2.3: Examples of RPM that show the effects of adding *noise* attributes. (Left) Position, Type, Size, and Color could vary freely as long as Number follows the rule. (Right) Position and Type in the inside group could vary freely.

that each grammar level could have multiple instantiations, *i.e.*, different categories or types. The Scene level could choose any available Structure, which consists of possibly multiple Components. Each Component branches into Layouts that links Entities. Attributes are appended to certain levels; for instance, (i) Number and Position are associated with Layout, and (ii) Type, Size, and Color are associated with Entity. Each attribute could take a value from a finite set. During sampling, both image structure and attribute values are sampled.

To increase the challenges and difficulties in the RAVEN dataset, we further append 2 types of *noise* attributes—Uniformity and Orientation—to Layout and Entity, respectively. Uniformity, set false, will not constrain Entities in a Layout to look the same, while Orientation allows an Entity to self-rotate. See Fig. 2.3 for the effects of the noise attributes.

This grammatical design of the image space allows the dataset to be very diverse and easily extendable. In this dataset, we manage to derive 7 configurations by combining different Structures, Components, and Layouts. Fig. 2.4 shows examples in each figure

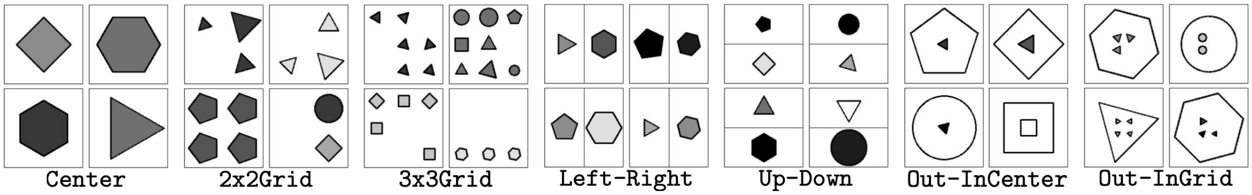


Figure 2.4: Examples of 7 different figure configurations in the proposed RAVEN dataset.

configuration.

2.3.2 Applying Rules

Carpenter *et al.* [CJS90] summarized that in the advanced RPM, rules were applied row-wise and could be grouped into 5 types. Unlike Berrett *et al.* [BHS18], we strictly follow Carpenter *et al.*'s description of RPM and implement all the rules, except that we merge `Distribute Two` into `Distribute Three`, as the former is essentially the latter with a null value in one of the attributes.

Specifically, we implement 4 types of rules in RAVEN: `Constant`, `Progression`, `Arithmetic`, and `Distribute Three`. Different from [BHS18], we add internal parameters to certain rules (*e.g.*, `Progression` could have increments or decrements of 1 or 2), resulting in a total of 8 distinct rule instantiations. Rules do not operate on the 2 noise attributes. As shown in Fig. 2.1 and Fig. 2.2, they are denoted as `[attribute:rule]` pairs.

To make the image space even more structured, we require each attribute to go over one rule and all `Entities` in the same `Component` to share the same set of rules, while different `Components` could vary.

Given the tree representation and the rules, we first prune the grammar tree such that all sub-trees satisfy the constraints imposed by the relations. We then sample from the tree and apply the rules to compose a row. Iterating the process three times yields a problem matrix.

2.3.3 Generating the Answer Set

To generate the answer set, we first derive the correct representation of the solution and then leverage the monotonicity of RPM constraints proposed by Wang and Su [WS15]. To break the correct relationships, we find an attribute that is constrained by a rule as described in Sec. 2.3.2 and vary it. By modifying only one attribute, we could greatly reduce the computation. Such modification also increases the difficulty of the problem, as it requires attention to subtle difference to tell an incorrect candidate from the correct one.

2.4 Comparison and Analysis

In this section, we compare RAVEN with the existing PGM, presenting its key features and some statistics in Sec. 2.4.1. In addition, we fill in two missing pieces in a desirable RPM dataset, *i.e.*, structure and hierarchy (Sec. 2.4.2), as well as the human performance (Sec. 2.4.3). We also show that RPM becomes trivial and could be solved instantly using a heuristics-based searching method (Sec. 2.4.4), given a symbolic representation of images and operations of rules.

2.4.1 Comparison with PGM

Tab. 2.1 summarizes several essential metrics of RAVEN and PGM. Although PGM is larger than RAVEN in terms of size, it is very limited in the average number of rules (**AvgRule**), rule instantiations (**RuleIns**), number of structures (**Struct**), and figure configurations (**FigConfig**). This contrast in PGM's gigantic size and limited diversity might disguise model fitting as a misleading reasoning ability, which is unlikely to generalize to other scenarios.

To avoid such an undesirable effect, we refrain from generating a dataset too large, even though our structured representation allows generation of a combinatorial number of problems. Rather, we set out to incorporate more rule instantiations (8), structures (4), and figure configurations (7) to make the dataset diverse (see Fig. 2.4 for examples). Note that an equal number of images for each figure configuration is generated in the RAVEN dataset.

2.4.2 Introduction of Structure

A distinctive feature of RAVEN is the introduction of the structural representation of the image space. Wang and Su [WS15] and Barrett *et al.* [BHS18] used plain logic and flat rule representations, respectively, resulting in no base of the structure to perform reasoning on. In contrast, we have in total $1,120,000$ structure annotations (**StructAnno**) in the form of parsed sentences in the dataset, pairing each problem instance with 16 sentences for both the matrix and the answer set. These representations derived from the A-SIG allow a new form of reasoning, *i.e.*, one that combines visual understanding and structure reasoning. As shown in [LF17, LFU10, LTF09] and our experiments in Sec. 2.6, incorporating structure into RPM problem solving could result in further performance improvement across different models.

	PGM [BHS18]	**RAVEN (Ours)**
AvgRule	1.37	6.29
RuleIns	5	8
Struct	1	4
FigConfig	3	7
StructAnno	0	1,120,000
HumanPerf		✓

Table 2.1: Comparison with the PGM dataset.

2.4.3 Human Performance Analysis

Another missing point in the previous work [BHS18] is the evaluation of human performance. To fill in the missing piece, we recruit human subjects consisting of college students from a subject pool maintained by the Department of Psychology to test their performance on a subset of representative samples in the dataset. In the experiments, human subjects were familiarized by solving problems with only one non-`Constant` rule in a fixed configuration. After the familiarization, subjects were asked to answer RPM problems with complex rule combinations, and their answers were recorded. Note that we deliberately included all figure configurations to measure generalization in the human performance and only "easily perceptible" examples were used in case certain subjects might have impaired perception. The results are reported in Tab. 2.2. The notable performance gap calls for further research into this problem. See Sec. 2.6 for detailed analysis and comparisons with vision models.

2.4.4 Heuristics-based Solver using Searching

We also find that the RPM could be essentially turned into a searching problem, given the symbolic representation of images and the access to rule operations as in [LF17, LFU10, LTF09]. Under such a setting, we could treat this problem as constraint satisfaction and develop a heuristics-based solver. The solver checks the number of satisfied constraints in each candidate answer and selects one with the highest score, resulting in perfect performance. Results are reported in Tab. 2.2. The optimality of the heuristic-based solver also verifies the well-formedness of RAVEN in the sense that there exists only one candidate that satisfies all constraints.

2.5 Dynamic Residual Tree for RPM

The image space of RPM is inherently structured and could be described using a symbolic language, as shown in [CJS90, LF17, LFU10, LTF09, Rav38]. To capture this characteristic and further improve the model performance on RPM, we propose a simple tree-structure neural module called Dynamic Residual Tree (DRT) that operates on the joint space of image understanding and structure reasoning. An example of DRT is shown in Fig. 2.5.

In the DRT, given a sentence S sampled from the A-SIG, usually represented as a serialized n-ary tree, we could first recover the tree structure. Note that the tree is **dynamically** generated following the sentence S, and each node in the tree comes with a label. With a structured tree representation ready, we could now consider assigning a neural computation operator to each tree node, similar to Tree-LSTM [TSM15]. To further simplify computation, we replace the LSTM cell [HS97] with a ReLU-activated [NH10] fully-connected layer f. In this way, nodes with a single child (leaf nodes or OR-production nodes) update the input features by

$$I = \text{ReLU}(f([I, w_n])), \tag{2.1}$$

where $[\cdot, \cdot]$ is the concatenation operation, I denotes the input features, and w_n the distributed representations of the node's label [MSC13, PSM14]. Nodes with multiple children (AND-production nodes) update input features by

$$I = \text{ReLU}\left(f\left(\left[\sum_c I_c, w_n\right]\right)\right), \tag{2.2}$$

where I_c denotes the features from its child c.

In summary, features from the lower layers are fed into the leaf nodes of DRT, gradually updated by Eq. (2.1) and Eq. (2.2) from bottom-up following the tree structure, and output to higher-level layers.

Inspired by [HZR16], we make DRT a **residual** module by adding the input and output

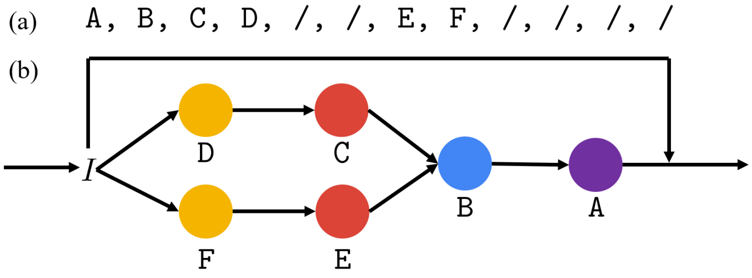


Figure 2.5: An example computation graph of DRT. (a) Given the serialized n-ary tree representation (pre-order traversal with / denoting end-of-branch), (b) a tree-structured computation graph is dynamically built. The input features are wired from bottom-up following the tree structure. The final output is the sum with the input, forming a residual module.

of DRT together, hence the name Dynamic Residual Tree (DRT)

$$I = \mathrm{DRT}(I, S) + I. \tag{2.3}$$

2.6 Experiments

2.6.1 Computer Vision Models

We adopt several representative models suitable for RPM and test their performances on RAVEN [BHS18, HZR16, KSH12, XCW15]. In summary, we test a simple sequential learning model (LSTM), a CNN backbone with an MLP head (CNN), a ResNet-based [HZR16] image classifier (ResNet), the recent relational WReN [BHS18], and all these models augmented with the proposed DRT.

LSTM The partially sequential nature of the RPM problem inspires us to borrow the power of sequential learning. Similar to ConvLSTM [XCW15], we feed each image feature extracted by a CNN into an LSTM network sequentially and pass the last hidden feature into a two-layer MLP to predict the final answer. In the DRT-augmented LSTM, *i.e.*, LSTM-DRT, we feed features of each image to a shared DRT before the final LSTM.

CNN We test a neural network model used in Hoshen and Werman [HW17]. In this model, a four-layer CNN for image feature extraction is connected to a two-layer MLP with a softmax layer to classify the answer. The CNN is interleaved with batch normalization [IS15] and ReLU non-linearity [NH10]. Random dropout [SHK14] is applied at the penultimate layer of MLP. In CNN-DRT, image features are passed to DRT before MLP.

ResNet Due to its surprising effectiveness in image feature extraction, we replace the feature extraction backbone in CNN with a ResNet [HZR16] in this model. We use a publicly available ResNet implementation, and the model is randomly initialized without pre-training. After testing several ResNet variants, we choose ResNet-18 for its good performance. The DRT extension and the training strategy are similar to those used in the CNN model.

WReN We follow the original paper [BHS18] in implementing the WReN. In this model, we first extract image features by a CNN. Each answer feature is then composed with each context image feature to form a set of ordered pairs. The order pairs are further fed to an MLP and summed. Finally, a softmax layer takes features from each candidate answer and makes a prediction. In WReN-DRT, we apply DRT on the extracted image features before the relational module.

For all DRT extensions, nodes in the same level share parameters and the representations for nodes' labels are fixed after initialization from corresponding 300-dimension GloVe vectors [PSM14]. Sentences used for assembling DRT could be either retrieved or learned by an encoder-decoder. Here we report results using retrieval.

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN	14.69%	13.09%	28.62%	28.27%	7.49%	6.34%	8.38%	10.56%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
LSTM+DRT	13.96%	14.29%	15.08%	14.09%	13.79%	13.24%	13.99%	13.29%
WReN+DRT	15.02%	15.38%	23.26%	29.51%	6.99%	8.43%	8.93%	12.35%
CNN+DRT	39.42%	37.30%	30.06%	34.57%	45.49%	45.54%	45.93%	37.54%
ResNet+DRT	**59.56%**	**58.08%**	**46.53%**	**50.40%**	**65.82%**	**67.11%**	**69.09%**	**60.11%**
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver*	100%	100%	100%	100%	100%	100%	100%	100%

Table 2.2: Testing accuracy of each model against human subjects and the solver. Acc denotes the mean accuracy of each model, while other columns show model accuracy on different figure configurations. L-R denotes Left-Right, U-D denotes Up-Down, O-IC denotes Out-InCenter, and O-IG denotes Out-InGrid. *Note that the perfect solver has access to rule operations and searches on the symbolic problem representation.

2.6.2 Experimental Setup

We split the RAVEN dataset into three parts, 6 folds for training, 2 folds for validation, and 2 folds for testing. We tune hyper-parameters on the validation set and report the model accuracy on the test set. For loss design, we treat the problem as a classification task and train all models with the cross-entropy loss. All the models are implemented in PyTorch [PGC17] and trained with ADAM [KB14] before early stopping or a maximum number of epochs is reached.

2.6.3 Performance Analysis

Tab. 2.2 shows the testing accuracy of each model trained on RAVEN, against the human performance and the heuristics-based solver. Neither human subjects nor the solver experiences an intensive training session, and the solver has access to the rule operations and searches the answer based on a symbolic representation of the problem. In contrast, all the computer vision models go over an extensive training session, but only on the training set.

In general, human subjects produce better testing accuracy on problems with simple figure configurations such as `Center`, while human performance reasonably deteriorates on problem instances with more objects such as `2x2Grid` and `3x3Grid`. Two interesting observations:

1. For figure configurations with multiple components, although each component in `Left-Right`, `Up-Down`, and `Out-InCenter` has only one object, making the reasoning similar to `Center` except that the two components are independent, human subjects become less accurate in selecting the correct answer.
2. Even if `Up-Down` could be regarded as a simple transpose of `Left-Right`, there exists some notable difference. Such effect is also implied by the "inversion effects" in cognition; for instance, inversion disrupts face perception, particularly sensitivity to spatial relations [CM09, LMM01].

In terms of model performance, a counter-intuitive result is: computer vision systems do not achieve the best accuracy across all other configurations in the seemingly easiest figure configuration for human subjects (`Center`). We further realize that the LSTM model and the WReN model perform only slightly better than random guess (12.5%). Such results contradicting to [BHS18] might be attributed to the diverse figure configurations in RAVEN. Unlike LSTM whose accuracy across different configurations is more or less uniform, WReN achieves higher accuracy on configurations consisting of multiple randomly distributed objects (`2x2Grid` and `3x3Grid`), with drastically degrading performance in configurations consisting of independent image components. This suggests WReN is biased to grid-like configurations (majority of PGM) but not others that require compositional reasoning (as in RAVEN). In contrast, a simple CNN model with MLP doubles the performance of WReN on RAVEN, with a tripled performance if the backbone is ResNet-18.

We observe a consistent performance improvement across different models after incorporating DRT, suggesting the effectiveness of the structure information in this visual reasoning problem. While the performance boost is only marginal in LSTM and WReN, we notice a marked accuracy increase in the CNN- and ResNet-based models (6.63% and 16.58% relative increase respectively). However, the performance gap between artificial vision systems and humans are still significant (up to 37% in `2x2Grid`), calling for further research to bridge the gap.

2.6.4 Effects of Auxiliary Training

Barrett *et al*. [BHS18] mentioned that training WReN with a fine-tuned auxiliary task could further give the model a 10% performance improvement. We also test the influence of auxiliary training on RAVEN. First, we test the effects of an auxiliary task to classify the rules and attributes on WReN and our best performing model ResNet+DRT. The setting is similar to [BHS18], where we perform an OR operation on a set of multi-hot vectors describing the rules and the attributes they apply to. The model is then tasked to both correctly find the answer and classify the rule set with its governing attributes. The final loss becomes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \beta \mathcal{L}_{\text{rule}}, \tag{2.4}$$

where $\mathcal{L}_{\text{target}}$ denotes the cross-entropy loss for the answer, $\mathcal{L}_{\text{rule}}$ the multi-label classification loss for the rule set, and β the balancing factor. We observe no performance change on WReN but a serious performance downgrade on ResNet+DRT (from 59.56% to 20.71%).

Since RAVEN comes with structure annotations, we further ask whether adding a structure prediction loss could help the model improve performance. To this end, we cast the experiment in a similar setting where we design a multi-hot vector describing the structure of each problem instance and train the model to minimize

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \alpha \mathcal{L}_{\text{struct}}, \tag{2.5}$$

where $\mathcal{L}_{\text{struct}}$ denotes the multi-label classification loss for the problem structure, and α the balancing factor. In this experiment, we observe a slight performance decrease in ResNet+DRT (from 59.56% to 56.86%). A similar effect is noticed on WReN (from 14.69% to 12.58%).

2.6.5 Test on Generalization

One interesting question we would like to ask is how a model trained well on one figure configuration performs on another similar figure configuration. This could be a measure of models' generalizability and compositional reasoning ability. Fortunately, RAVEN naturally provides us with a test bed. To do this, we first identify several related configuration regimes:

- Train on `Center` and test on `Left-Right`, `Up-Down`, and `Out-InCenter`. This setting directly challenges the compositional reasoning ability of the model as it requires the model to generalize the rules learned in a single-component configuration to configurations with multiple independent but similar components.
- Train on `Left-Right` and test on `Up-Down`, and vice-versa. Note that for `Left-Right` and `Up-Down`, one could be regarded as a transpose of another. Thus, the test could measure whether the model simply memorizes the pattern in one configuration.
- Train on `2x2Grid` and test on `3x3Grid`, and vice-versa. Both configurations involve multi-object interactions. Therefore the test could measure the generalization when the number of objects changes.

The following results are all reported using the best performing model, *i.e.*, ResNet+DRT.

Tabs. 2.3 to 2.5 show the result of our model generalization test. We observe:

`Center`	`Left-Right`	`Up-Down`	`Out-InCenter`
51.87%	40.03%	35.46%	38.84%

Table 2.3: Generalization test. The model is trained on `Center` and tested on three other configurations.

	`Left-Right`	`Up-Down`
`Left-Right`	41.07%	38.10%
`Up-Down`	39.48%	43.60%

Table 2.4: Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

- The model dedicated to a single figure configuration does not achieve better test accuracy than one trained on all configurations together. This effect justifies the importance of the diversity of RAVEN, showing that increasing the number of figure configurations could actually improve the model performance.
- Tab. 2.3 also implies that a certain level of compositional reasoning, though weak, exists in the model, as the three other configurations could be regarded as a multi-component composition of `Center`.
- In Tab. 2.4, we observe no major differences in terms of test accuracy. This suggests that the model could successfully transfer the knowledge learned in a scenario to a very similar counterpart, when one configuration is the transpose of another.
- From Tab. 2.5, we notice that the model trained on `3x3Grid` could generalize to `2x2Grid` with only minor difference from the one dedicated to `2x2Grid`. This could be attributed to the fact that in the `3x3Grid` configuration, there could be instances with object distribution similar to that in `2x2Grid`, but not vice versa.

	2x2Grid	3x3Grid
2x2Grid	40.93%	38.69%
3x3Grid	39.14%	43.72%

Table 2.5: Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

2.7 Conclusion

We present a new dataset for Relational and Analogical Visual Reasoning in the context of Raven's Progressive Matrices (RPM), called RAVEN. Unlike previous work, we apply a systematic and structured tool, *i.e.*, Attributed Stochastic Image Grammar (A-SIG), to generate the dataset, such that every problem instance comes with rich annotations. This tool also makes RAVEN diverse and easily extendable. One distinguishing feature that tells apart RAVEN from other work is the introduction of the structure. We also recruit quality human subjects to benchmark human performance on the RAVEN dataset. These aspects fill two important missing points in previous works.

We further propose a novel neural module called Dynamic Residual Tree (DRT) that leverages the structure annotations for each problem. Extensive experiments show that models augmented with DRT enjoy consistent performance improvement, suggesting the effectiveness of using structure information in solving RPM. However, the difference between machine algorithms and humans clearly manifests itself in the notable performance gap, even in an unfair situation where machines experience an intensive training session while humans do not. We also realize that auxiliary tasks do not help performance on RAVEN. The generalization test shows the importance of diversity of the dataset, and also indicates current computer vision methods do exhibit a certain level of reasoning ability, though weak.

The entire work still leaves us many mysteries. Humans seem to apply a combination of the top-down and bottom-up method in solving RPM. How could we incorporate this into a model? What is the correct way of formulating visual reasoning? Is it model fitting? Is deep learning the ultimate way to visual reasoning? If not, how could we revise the models? If yes, how could we improve the models?

Finally, we hope these unresolved questions would call for attention into this challenging problem.

CHAPTER 3

Data-driven Few-shot Concept Induction by Contrasting

"Thinking in pictures," [Gra06] *i.e.*, spatial-temporal reasoning, effortless and instantaneous for humans, is believed to be a significant ability to perform logical induction and a crucial factor in the intellectual history of technology development. Modern AI, fueled by massive datasets, deeper models, and mighty computation, has come to a stage where (super-)human-level performances are observed in certain specific tasks. However, current AI's ability in "thinking in pictures" is still far lacking behind. In this work, we study how to improve machines' reasoning ability on one challenging task of this kind: RPM. Specifically, we borrow the very idea of "contrast effects" from the field of psychology, cognition, and education to design and train a permutation-invariant model. Inspired by cognitive studies, we equip our model with a simple inference module that is jointly trained with the perception backbone. Combining all the elements, we propose the *Contrastive Perceptual Inference* network (CoPINet) and empirically demonstrate that CoPINet sets the new state-of-the-art for permutation-invariant models on two major datasets. We conclude that spatial-temporal reasoning depends on envisaging the possibilities consistent with the relations between objects and can be solved from pixel-level inputs.

3.1 Introduction

Among the broad spectrum of computer vision tasks are ones where dramatic progress has been witnessed, especially those involving visual information retrieval [KSH12, HZR16, RDG16, RHG15]. Significant improvement has also manifested itself in tasks associating visual and linguistic understanding [AAL15, JHM17a, JHM17b, HAR17]. However, it was only until recently that the research community started to re-investigate tasks relying heavily on the ability of "thinking in pictures" with modern AI approaches [Gra06, Arn69, Gal83], particularly spatial-temporal inductive reasoning [ZGJ19, HSB19, BHS18]; this line of work primarily focuses on Raven's Progressive Matrices (RPM) [Rav36, RC98]. It is believed that RPM is closely related to real intelligence [CJS90], diagnostic of abstract and structural reasoning ability [EKM84], and characterizes *fluid intelligence* [Spe27, Spe23, Hof95, JBJ08]. In such a test, subjects are provided with two rows of figures following certain *unknown* rules and asked to pick the correct answer from the choices that would best complete the third row with a missing entry; see Fig. 3.1(a) for an example. As shown in early works [ZGJ19, BHS18], despite the fact that *visual elements* are relatively straightforward, there is still a notable performance gap between human and machine *visual reasoning* in this challenging task.

One missing ingredient that may result in this performance gap is a proper form of contrasting mechanism. Originated from perceptual learning [GG55, Gib14], it is well established in the field of psychology and education [CH89, GG01, HDW09, GP92, HIO11] that teaching new concepts by comparing with noisy examples is quite effective. Smith *et al*. [SG14] summarize that comparing cases facilitates transfer learning and problem-solving, as well as the ability to learn relational categories. Gentner *et al*. [Gen83] in his structure-mapping theory points out that learners generate a structure alignment between two representation when they compare two cases. A more recent study from Schwartz *et al*. [SCO11] also shows that contrasting cases help foster an appreciation of a deep understanding of concepts.

We argue that such a *contrast effect* [Bow61], found in both humans and animals [Mey51, SH56, SV78, Law57, Ams62], is essential to machines' reasoning ability as well. With access to how the data is generated, a recent attempt [HSB19] finds that models demonstrate better generalizability if the choice of data and the manner in which it is presented to the model are made "contrastive." In this paper, we try to address a more direct and challenging question, *independent* of how the data is generated: how to incorporate an explicit contrasting mechanism during model *training* in order to improve machines' reasoning ability? Specifically, we come up with two levels of contrast in our model: a novel contrast module and a new contrast loss. At the model level, we design a permutation-invariant contrast module that summarizes the common features and distinguishes each candidate by projecting it onto its residual on the common feature space. At the objective level, we leverage ideas in contrastive estimation [GH10, SE05, DL17] and propose a variant of NCE loss.

Another reason why RPM is challenging for existing machine reasoning systems could be attributed to the demanding nature of the *interplay* between perception and inference. Carpenter *et al*. [CJS90] postulate that a proper understanding of one RPM instance requires not only an accurate encoding of individual elements and their visual attributes but also the correct induction of the hidden rules. In other words, to solve RPM, machine reasoning systems are expected to be equipped with *both* perception and inference subsystems; lacking either component would only result in a sub-optimal solution. While existing work primarily focuses on perception, we propose to bridge this gap with a simple inference module *jointly* trained with the perception backbone; specifically, the inference module reasons about which category the current problem instance falls into. Instead of training the inference module to predict the ground-truth category, we borrow the basis learning idea from [WSG10] and jointly learn the inference subsystem with perception. This basis formulation could also be regarded as a hidden variable and trained using a log probability estimate.

Furthermore, we hope to make a critical improvement to the model design such that it is truly *permutation-invariant*. The invariance is mandatory, as an ideal RPM solver should not change the representation simply because the rows or columns of answer candidates are swapped or the order of the choices alters. This characteristic is an essential trait missed by all recent works [ZGJ19, BHS18]. Specifically, Zhang *et al.* [ZGJ19] stack all choices in the channel dimension and feed it into the network in one pass. Barrett *et al.* [BHS18] add additional positional tagging to their WReN. Both of them *explicitly* make models permutation-sensitive. We notice in our experiments that removing the positional tagging in WReN decreases the performance by 28%, indicating that the model bypasses the intrinsic complexity of RPM by remembering the positional association. Making the model permutation-invariant also shifts the problem from classification to ranking.

Combining contrasting, perceptual inference, and permutation invariance, we propose the *Contrastive Perceptual Inference* network (CoPINet). To verify its effectiveness, we conduct comprehensive experiments on two major datasets: the RAVEN dataset [ZGJ19] and the PGM dataset [BHS18]. Empirical studies show that our model achieves human-level performance on RAVEN and a new record on PGM, setting new state-of-the-art for permutation-invariant models on the two datasets. Further ablation on RAVEN and PGM reveals how each component contributes to performance improvement. We also investigate how the model performance varies under different sizes of datasets, as a step towards an ideal machine reasoning system capable of low-shot learning.

This paper makes four major contributions:

- We introduce two levels of contrast to improve machines' reasoning ability in RPM. At the model level, we design a contrast module that aggregates common features and projects each candidate to its residual. At the objective level, we use an NCE loss variant instead of the cross-entropy to encourage contrast effects.
- Inspired by Carpenter *et al.* [CJS90], we incorporate an inference module to learn with the perception backbone jointly. Instead of using ground-truth, we regularize it with a fixed number of bases.
- We make our model permutation-invariant in terms of swapped rows or columns and shuf-

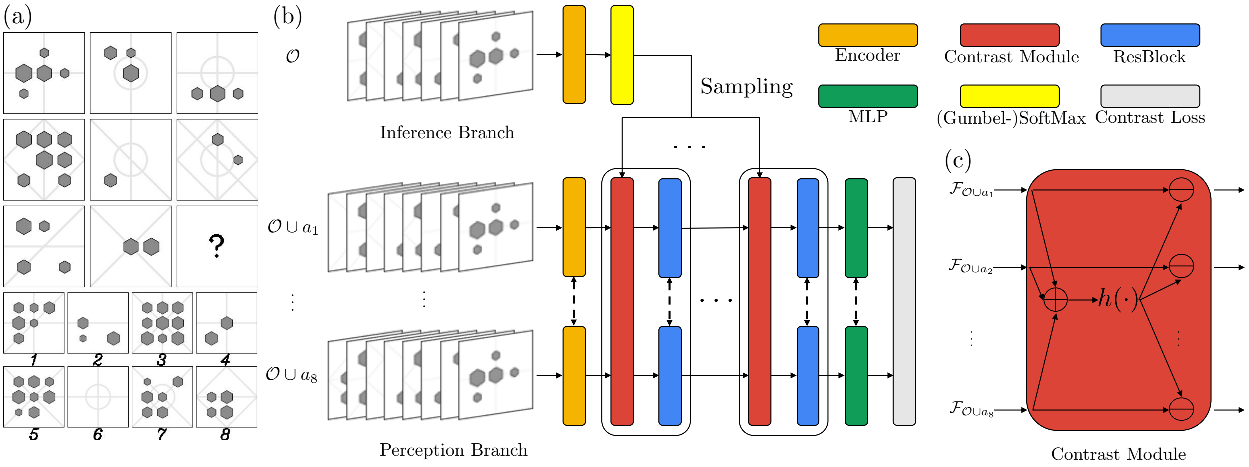


Figure 3.1: (a) An example of RPM. The hidden rule(s) in this problem can be denoted as $\{[\text{OR}, \text{line}, \text{type}]\}$, where an OR operation is applied to the type attribute of all lines, following the notations in Barrett *et al.* [BHS18]. It is further noted that the OR operation is applied row-wise, and there is only one choice that satisfies the row-wise OR constraint. Hence the correct answer should be 5. (b) The proposed CoPINet architecture. Given a RPM problem, the inference branch samples a most likely rule for each attribute based only on the context $\mathcal{O}$ of the problem. Sampled rules are transformed and fed into each contrast module in the perception branch. Note that the combination of the contrast module and the residual block can be repeated. Dashed lines indicate that parameters are shared among the modules. (c) A sketch of the contrast module.

fied answer candidates, shifting the previous view of RPM from classification to ranking.

- Combining ideas above, we propose CoPINet that sets new state-of-the-art on two major datasets.

3.2 Related Work

Contrastive Learning Teaching concepts by comparing cases, or contrasting, has proven effective in both human learning and machine learning. Gentner *et al.* [Gen83] postulates that human's learning-by-comparison process is a structural mapping and alignment process.

A later article [GM94] firmly supports this conjecture and shows finding the individual difference is easier for humans when similar items are compared. Recently, Smith *et al.* [SG14] conclude that learning by comparing two contrastive cases facilitates the distinction between two complex interrelated relational concepts. Evidence in educational research further strengthens the importance of contrasting—quantitative structure of empirical phenomena is less demanding to learn when contrasting cases are used [SCO11, CSS10, SM04]. All the literature calls for a similar treatment of contrast in machine learning. While techniques from [CHL05, WS09, WG15] are based on triplet loss using max margin to separate positive and negative samples, negative contrastive samples and negative sampling are proposed for language modeling [SE05] and word embedding [MSC13, KZS15], respectively. Gutmann *et al.* [GH10] discuss a general learning framework called Noise-Contrastive Estimation (NCE) for estimating parameters by taking noise samples into consideration, which Dai *et al.* [DL17] follow to learn an effective image captioning model. A recent work [HSB19] leverages contrastive learning in RPM; however, it focuses on data presentation while leaving the question of modeling and learning unanswered.

Computational Models on RPM The cognitive science community is the first to investigate RPM with computational models. Assuming access to a perfect state representation, structure-mapping theory [Gen83] and the high-level perception theory of analogy [CFH92, Mit93] are designed with heuristics to solve the RPM problem at a symbolic level [CJS90, LF17, LFU10, LTF09]. Another stream of research approaches the problem by measuring the image similarity with hand-crafted state representations [LLG12, MG14, MKG14, MSD18, SG18a]. More recently, end-to-end data-driven methods with raw image input are proposed [ZGJ19, HSB19, BHS18, WS15]. Want *et al.* [WS15] introduce an automatic RPM generation method. Battett *et al.* [BHS18] release the first large-scale RPM dataset and present a relational model [SRB17] designed for it. Steenbrugge *et al.* [SLV18] propose a pretrained β-VAE to improve the generalization performance of models on RPM.

Zhang *et al.* [ZGJ19] provide another dataset with structural annotations using stochastic image grammar [ZM07, PZ15, WXZ07]. Hill *et al.* [HSB19] take a different approach and study how data presentation affects learning.

3.3 Learning Perceptual Inference by Contrasting

The task of RPM can be formally defined as: given a list of observed images $\mathcal{O} = \{o_i\}_{i=1}^{8}$, forming a 3 × 3 matrix with a final missing element, a solver aims to find an answer $a_\star$ from an *unordered* set of choices $\mathcal{A} = \{a_i\}_{i=1}^{8}$ to best complete the matrix. Permutation invariance is a unique property for RPM problems: (1) According to [CJS90], the same set of rules is applied either row-wise or column-wise. Therefore, swapping the first two rows or columns should not affect how one solves the problem. (2) In any multi-choice task, changing the order of answer candidates should not affect how one solves the problem either. These properties require us to use a permutation-invariant encoder and reformulate the problem from a typical classification problem into a ranking problem. Formally, in a probabilistic formulation, we seek to find a model such that

$$p(a_\star|\mathcal{O}) \geqslant p(a'|\mathcal{O}), \quad \forall a' \in \mathcal{A}, a' \neq a_\star, \tag{3.1}$$

where the probability is invariant when rows or columns in $\mathcal{O}$ are swapped. This formulation also calls for a model that produces a density estimation for each choice, regardless of its order in $\mathcal{A}$. To that end, we model the probability with a neural network equipped with a permutation-invariant encoder for each observation-candidate pair $f(\mathcal{O} \cup a)$. However, we argue such a purely perceptive system is far from sufficient without contrasting and perceptual inference.

3.3.1 Contrasting

To provide the reasoning system with a mechanism of contrasting, we propose to explicitly build two levels of contrast: model-level contrast and objective-level contrast.

3.3.1.1 Model-level Contrast

As the central notion of contrast is comparing cases [SG14, SCO11, CSS10, SM04], we propose an explicit model-level contrasting mechanism in the following form,

$$\text{Contrast}(\mathcal{F}_{\mathcal{O}\cup a}) = \mathcal{F}_{\mathcal{O}\cup a} - h\left(\sum_{a'\in\mathcal{A}} \mathcal{F}_{\mathcal{O}\cup a'}\right), \tag{3.2}$$

where $\mathcal{F}$ denotes features of a specific combination and $h(\cdot)$ summarizes the common features in all candidate answers. In our experiments, $h(\cdot)$ is a composition of BatchNorm [IS15] and Conv.

Intuitively, this explicit contrasting computation enables a reasoning system to tell distinguishing features for each candidate in terms of fitting and following the rules hidden among all panels in the incomplete matrix. The philosophy behind this design is to constrain the functional form of the model to capture both the commonality and the difference in each instance. It is expected that the very inductive bias on comparing similarity and distinctness is baked into the entire reasoning system such that learning in the challenging task becomes easier.

In a generalized setting, each $\mathcal{O} \cup a$ could be abstracted out as an object. Then the design becomes a general contrast module, where each object is distinguished by comparing with the common features extracted from an object set.

We further note that the contrasting computation can be encapsulated into a single neural module and repeated: the addition and transformation are shared and the subtraction is performed on each individual element. See Fig. 3.1(c) for a sketch of the contrast module. After such operations, permutation invariance of a model will not be broken.

3.3.1.2 Objective-level Contrast

To further enforce the contrast effects, we propose to use an NCE variant rather than the cross-entropy loss commonly used in previous works [ZGJ19, BHS18]. While there are several ways to model the probability in Eq. (3.1), we use a Gibbs distribution in this work:

$$p(a|\mathcal{O}) = \frac{1}{Z}\exp(f(\mathcal{O} \cup a)), \tag{3.3}$$

where Z is the partition function, and our model $f(\cdot)$ corresponds to the negative potential function. Note that such a distribution has been widely adopted in image generation models [ZWM98, WXL18, XLZ16].

In this case, we can take the log of both sides in Eq. (3.1) and rearrange terms:

$$\log p(a_\star|\mathcal{O}) - \log p(a'|\mathcal{O}) = f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a') \geqslant 0, \quad \forall a' \in \mathcal{A}, a' \neq a_\star. \tag{3.4}$$

This formulation could potentially lead to a max margin loss. However, we notice in our preliminary experiments that max margin is not sufficient; we realize it is inferior to make the negative potential of the wrong choices only *slightly lower*. Instead, we would like to further push the difference to *infinity*. To do that, we leverage the *sigmoid* function $\sigma(\cdot)$ and train the model, such that:

$$f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a') \to \infty \iff \sigma(f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a')) \to 1, \forall a' \in \mathcal{A}, a' \neq a_\star. \tag{3.5}$$

However, we notice that the relative difference of negative potential is still problematic. We hypothesize this deficiency is due to the lack of a baseline—without such a regularization, the negative potential of wrong choices could still be very high, resulting in difficulties in learning the negative potential of the correct answer. To this end, we modify Eq. (3.5) into its sufficient conditions:

$$f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star) \to \infty \iff \sigma(f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star)) \to 1 \tag{3.6}$$

$$f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a') \to -\infty \iff \sigma(f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a')) \to 0, \tag{3.7}$$

where $b(\cdot)$ is a fixed baseline function and $a' \in \mathcal{A}, a' \neq a_\star$. For implementation, $b(\cdot)$ could be either a randomly initialized network or a constant. Since the two settings do not produce significantly different results in our preliminary experiments, we set $b(\cdot)$ to be a constant to reduce computation.

We then optimize the network to maximize the following objective as done in [GH10]:

$$\ell = \log(\sigma(f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star))) + \sum_{a' \in \mathcal{A}, a' \neq a_\star} \log(1 - \sigma(f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a'))). \tag{3.8}$$

Connection to NCE If we treat the baseline as the negative potential of a fixed noise model of the same Gibbs form and ignore the difference between the partition functions, Eq. (3.6) and Eq. (3.7) become the G function used in NCE [GH10]. But unlike NCE, we do not need to multiply the size ratio in the sigmoid function [DL17].

3.3.2 Perceptual Inference

As indicated in Zhang *et al.* [CJS90], a mere perceptive model for RPM is arguably not enough. Therefore, we propose to incorporate a simple inference subsystem into the model: the inference branch should be responsible for inferring the hidden rules in the problem. Specifically, we assume there are at most N attributes in each problem, each of which is subject to the governance of one of M rules. Then hidden rules $\mathcal{T}$ in one problem instance can be decomposed into

$$p(\mathcal{T}|\mathcal{O}) = \prod_{i=1}^{N} p(t_i|\mathcal{O}), \tag{3.9}$$

where $t_i = 1 \ldots M$ denotes the rule type on attribute n_i. For the actual form of the probability of rules on each attribute, we propose to model it using a multinomial distribution. This assumption is consistent with the way datasets are usually generated [ZGJ19, BHS18, WS15]: one rule is independently picked from the rule set for each attribute. In this way, each rule could also be regarded as a basis in a rule dictionary and jointly learned, as done in active basis [WSG10] or word embedding [MSC13, PSM14].

If we treat rules as hidden variables, the log probability in Eq. (3.4) can be decomposed into

$$\log p(a|\mathcal{O}) = \log \sum_{\mathcal{T}} p(a|\mathcal{T}, \mathcal{O}) p(\mathcal{T}|\mathcal{O}) = \log \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T}|\mathcal{O})}[p(a|\mathcal{T}, \mathcal{O})]. \tag{3.10}$$

Note that writing the summation in the form of expectation affords sampling algorithms, which can be done on each individual attribute due to the independence assumption.

In addition, if we model $p(\mathcal{T}|\mathcal{O})$ as an inference branch $g(\cdot)$ and sample only once from it, the model can be modified into $f(\mathcal{O} \cup a, \hat{\mathcal{T}})$ with $\hat{\mathcal{T}}$ sampled from $g(\mathcal{O})$. Following the same derivation above, we now optimize the new objective:

$$\ell = \log(\sigma(f(\mathcal{O} \cup a_\star, \hat{\mathcal{T}}) - b(\mathcal{O} \cup a_\star))) + \sum_{a' \in \mathcal{A}, a' \neq a_\star} \log(1 - \sigma(f(\mathcal{O} \cup a', \hat{\mathcal{T}}) - b(\mathcal{O} \cup a'))). \tag{3.11}$$

To sample from a multinomial, we could either use hard sampling like Gumbel-SoftMax [JGP16, MMT16] or a soft one by taking expectation. We do not observe significant difference between the two settings.

The expectation in Eq. (3.10) is proposed primarily to make the computation of the exact log probability controllable and tractable: while the full summation requires $O(M^N)$ passes of the model, a Monte Carlo approximation of it could be calculated in $O(1)$ time. We also note that if $p(\mathcal{T}|\mathcal{O})$ is highly peaked (*e.g.*, ground truth), the Monte Carlo estimate could be accurate as well. Despite the fact that we only sample once from an inference branch to reduce computation, we find in practice the Monte Carlo estimate works quite well.

3.3.3 Architecture

Combining contrasting, perceptual inference, and permutation invariance, we propose a new network architecture to solve the challenging RPM problem, named *Contrastive Perceptual Inference* network (CoPINet). The perception branch is composed of a common feature encoder and shared interweaving contrast modules and residual blocks [HZR16]. The encoder first extracts image features independently for each panel and sum ones in the corresponding rows and columns before the final transformation into a latent space. The inference branch consists of the same encoder and a (Gumbel-)SoftMax output layer. The sampled results will be transformed and concatenated channel-wise into the summation in Eq. (3.2). In our implementation, we prepend each residual block with a contrast module; such a combination can be repeated while keeping the network permutation-invariant. The network finally uses an MLP to produce a negative potential for each observation and candidate pair and is trained using Eq. (3.11); see Fig. 3.1(b) for a graphical illustration of the entire CoPINet architecture.

3.4 Experiments

3.4.1 Experimental Setup

We verify the effectiveness of our models on two major RPM datasets: RAVEN [ZGJ19] and PGM [BHS18]. Across all experiments, we train models on the training set, tune hyper-parameters on the validation set, and report the final results on the test set. All of the models are implemented in PyTorch [PGC17] and optimized using ADAM [KB14]. While a good performance of WReN [BHS18] and ResNet+DRT [ZGJ19] relies on external supervision, such as rule specifications and structural annotations, the proposed model achieves better performance with only $\mathcal{O}$, $\mathcal{A}$, and $a_\star$. Models are trained on servers with four Nvidia RTX Titans. For the WReN model, we use a public implementation that reproduces results in [BHS18]. We implement our models in PyTorch [PGC17] and optimize using ADAM [KB14]. During training, we perform early-stop based on validation loss. We use the same network architecture and hyper-parameters in both RAVEN and PGM experiments.

3.4.2 Results on RAVEN

There are $70,000$ problems in the RAVEN dataset [ZGJ19], equally distributed in 7 figure configurations. In each configuration, the dataset is randomly split into 6 folds for training, 2 folds for validation, and 2 folds for testing. We compare our model with several simple baselines (LSTM [HS97], CNN [HW17], and vanilla ResNet [HZR16]) and two strong baselines (WReN [BHS18] and ResNet+DRT [ZGJ19]). Model performance is measured by accuracy.

General Performance on RAVEN In this experiment, we train the models on all $42,000$ training samples and measure how they perform on the test set. The first part of Tab. 3.1 shows the testing accuracy of all models. We also retrieve the performance of humans and a solver with perfect information from [ZGJ19] for comparison. As shown in the table, the proposed model CoPINet achieves the best performance among all the models we test. For the relational model WReN proposed in [BHS18], we run the tests on a permutation-invariant version, *i.e.*, one without positional tagging (NoTag), and tune the model also to minimize an auxiliary loss (Aux) [BHS18]. While the auxiliary loss could boost the performance of WReN as we will show later in the ablation study, we do not observe similar effects on CoPINet. As indicated in the detailed comparisons in Tab. 3.1, WReN is biased towards images of grid configurations and does poorly on ones demanding compositional reasoning, *i.e.*, ones with independent components. We further note that compared to previously proposed models (WReN [BHS18] and ResNet+DRT [ZGJ19]), CoPINet does not require additional information such as structural annotations and meta targets and still shows human-level performance in this task. When comparing the performance of CoPINet and human on specific figure configurations, we notice that CoPINet is inferior in learning samples of grid-like compositionality but efficient in distinguishing images consisting of multiple components, implying the efficiency of the contrasting mechanism.

Ablation Study One problem of particular interest in building CoPINet is how each component contributes to performance improvement. To answer this question, we measure model accuracy by gradually removing each construct in CoPINet, *i.e.*, the perceptual inference branch, the contrast loss, and the contrast module. In the second part of Tab. 3.1, we show the results of ablation on CoPINet. Both the full model (CoPINet) and the one without the perceptual inference branch (CoPINet-Contrast-CL) could achieve human-level performance, with the latter slightly inferior to the former. If we further replace the contrast loss with the cross-entropy loss (CoPINet-Contrast-XE), we observe a noticeable performance decrease of around 4%, verifying the effectiveness of the contrast loss. A catastrophic performance downgrade of 66% is observed if we remove the contrast module, leaving only the

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN-NoTag-Aux	17.62%	17.66%	29.02%	34.67%	7.69%	7.89%	12.30%	13.94%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
ResNet+DRT	59.56%	58.08%	46.53%	50.40%	65.82%	67.11%	69.09%	60.11%
CoPINet	**91.42%**	**95.05%**	**77.45%**	**78.85%**	**99.10%**	**99.65%**	**98.50%**	**91.35%**
WReN-NoTag-NoAux	15.07%	12.30%	28.62%	29.22%	7.20%	6.55%	8.33%	13.10%
WReN-Tag-NoAux	17.94%	15.38%	29.81%	32.94%	11.06%	10.96%	11.06%	14.54%
WReN-Tag-Aux	33.97%	58.38%	38.89%	37.70%	21.58%	19.74%	38.84%	22.57%
CoPINet-Backbone-XE	20.75%	24.00%	23.25%	23.05%	15.00%	13.90%	21.25%	24.80%
CoPINet-Contrast-XE	86.16%	87.25%	71.05%	74.45%	97.25%	97.05%	93.20%	82.90%
CoPINet-Contrast-CL	90.04%	94.30%	74.00%	76.85%	99.05%	99.35%	98.00%	88.70%
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver	100%	100%	100%	100%	100%	100%	100%	100%

Table 3.1: Testing accuracy of models on RAVEN. Acc denotes the mean accuracy of each model. Same as in [ZGJ19], L-R denotes the Left-Right configuration, U-D Up-Down, O-IC Out-InCenter, and O-IG Out-InGrid.

network backbone (CoPINet-Backbone-XE). This drastic performance gap shows that the functional constraint on modeling an explicit contrasting mechanism is arguably a crucial factor in machines' reasoning ability as well as in humans'. The ablation study shows that all the three proposed constructs, especially the contrast module, are critical to the performance of CoPINet. We also study how the requirement of permutation invariance and auxiliary training affect the previously proposed WReN. As shown in Tab. 3.1, sacrificing the permutation invariance (Tag) provides the model a huge upgrade during auxiliary training (Aux), compared to the one without tagging (NoTag) and auxiliary loss (NoAux). This effect becomes even more significant on the PGM dataset, as we will show in Sec. 3.4.3.

Dataset Size and Performance Even though CoPINet surpasses human performance on RAVEN, this competition is inherently unfair, as the human subjects in this study never experience such an intensive training session as our model does. To make the comparison fairer and also as a step towards a model capable of human learning efficiency, we further measure how the model performance changes as the training set size shrinks. To this end, we train our CoPINet on subsets of the full RAVEN training set and test it on the full test set. As shown on Tab. 3.2 and Fig. 3.2, the model performance varies roughly log-linearly with the training set size. One surprising observation is: with only half of the amount of the data, we could already achieve human-level performance. On a training set 16× smaller, CoPINet outperforms all previous models. And on a subset 64× smaller, CoPINet already outshines WReN.

3.4.3 Results on PGM

We use the neutral regime of the PGM dataset for model evaluation due to its diversity and richness in relationships, objects, and attributes. This split of the dataset has in total 1.42 million samples, with 1.2 million for training, $2,000$ for validation, and $200,000$ for testing. We train the models on the training set, tune the hyperparameters on the validation set,

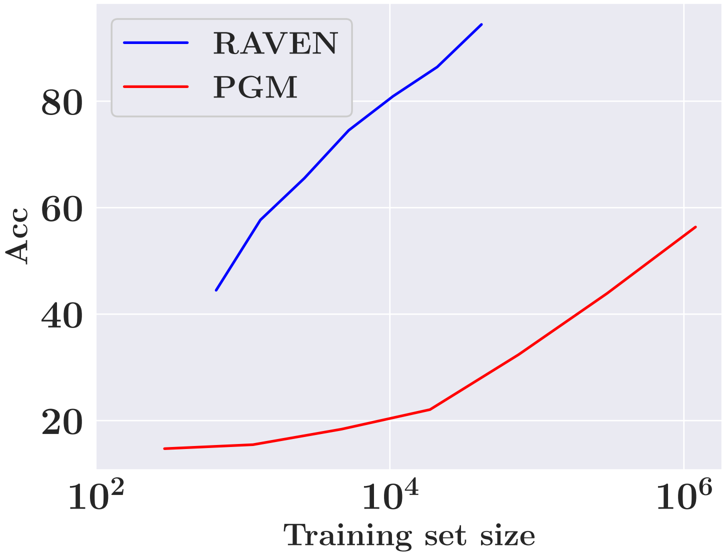


Figure 3.2: CoPINet on RAVEN and PGM as the training set size shrinks.

Training set size	Acc
658	44.48%
1,316	57.69%
2,625	65.55%
5,250	74.53%
10,500	80.92%
21,000	86.43%

Table 3.2: Model performance under different training set sizes on RAVEN dataset. The full training set has 42,000 samples.

and evaluate the performance on the test set. We compare our models with baselines set up in [BHS18], *i.e.*, LSTM, CNN, ResNet, Wild-ResNet, and WReN. As ResNet+DRT proposed in [ZGJ19] requires structural annotations not available in PGM, we are unable to measure its performance. Again, all performance is measured by accuracy. Due to the lack of further stratification on this training regime, we only report the final mean accuracy.

Training set size	Acc
293	14.73%
1,172	15.48%
4,688	18.39%
18,750	22.07%
75,000	32.39%
300,000	43.89%

Table 3.3: Model performance under different training set sizes on PGM dataset. The full training set has 1.2 million samples.

Method	CNN	LSTM	ResNet	Wild-ResNet	WReN-NoTag-Aux	CoPINet
Acc	33.00%	35.80%	42.00%	48.00%	49.10%	**56.37**%

Table 3.4: Testing accuracy of models on PGM. Acc denotes the mean accuracy of each model.

General Performance on PGM In this experiment, we train the models on all 1.2 million training samples and report performance on the entire test set. As shown in Tab. 3.4, CoPINet achieves the best performance among all permutation-invariant models, setting a new state-of-the-art on this dataset. Similar to the setting in RAVEN, we make the previously proposed WReN permutation-invariant by removing the positional tagging (NoTag) and train it with both cross-entropy loss and auxiliary loss (Aux) [BHS18]. The auxiliary loss could boost the performance of WReN. However, in coherence with the study on RAVEN and a previous work [ZGJ19], we notice that the auxiliary loss does not help our CoPINet. It is worth noting that while WReN demands additional training supervision from meta targets to reach the performance, CoPINet only requires basic annotations of ground truth indices $a_\star$ and achieves better results.

Ablation Study We perform ablation studies on both WReN and CoPINet to see how the requirement of permutation invariance affects WReN and how each module in CoPINet contributes to its superior performance. The notations are the same as those used in the ablation study for RAVEN. As shown in the first part of Tab. 3.5, adding a proper auxiliary loss does provide WReN a 10% performance boost. However, additional supervision is required. Making the model permutation-sensitive gives the model a significant benefit by up to a 28% accuracy increase; however, it also indicates that WReN learns to shortcut the solutions by coding the positional association, instead of truly understanding the differences among distinctive choices and their potential effects on the compatibility of the entire matrix. The second part of Tab. 3.5 demonstrates how each construct contributes to the performance improvement of CoPINet on PGM. Despite the smaller enhancement of the contrast loss compared to that in RAVEN, the upgrade from the contrast module for PGM is still significant, and the perceptual inference branch keeps raising the final performance. In accordance with the ablation study on the RAVEN dataset, we show that all the proposed components contribute to the final performance increase.

Method	WReN-NoTag-NoAux	WReN-NoTag-Aux	WReN-Tag-NoAux	WReN-Tag-Aux
Acc	39.25%	49.10%	62.45%	77.94%
Method	CoPINet-Backbone-XE	CoPINet-Contrast-XE	CoPINet-Contrast-CL	CoPINet
Acc	42.10%	51.04%	54.19%	56.37%

Table 3.5: Ablation study on PGM.

Dataset Size and Performance Motivated by the idea of fairer comparison and low-shot reasoning, we also measure how the performance of the proposed CoPINet changes as the training set size of PGM varies. Specifically, we train CoPINet on subsets of the PGM training set and test it on the entire test set. As shown in Tab. 3.3 and Fig. 3.2, CoPINet performance on PGM varies roughly log-exponentially with respect to the training set size.

We further note that when trained on a 16× smaller dataset, CoPINet already achieves results similar to CNN and LSTM.

3.5 Conclusion and Discussion

In this work, we aim to improve machines' reasoning ability in "thinking in pictures" by jointly learning perception and inference via contrasting. Specifically, we introduce the contrast module, the contrast loss, and the joint system of perceptual inference. We also require our model to be permutation-invariant. In a typical and challenging task of this kind, Raven's Progressive Matrices (RPM), we demonstrate that our proposed model—*Contrastive Perceptual Inference* network (CoPINet)—achieves the new state-of-the-art for permutation-invariant models on two major RPM datasets. Further ablation studies show that all the three proposed components are effective towards improving the final results, especially the contrast module. It also shows that the permutation invariance forces the model to understand the effects of different choices on the compatibility of an entire RPM matrix, rather than remembering the positional association and shortcutting the solutions.

While it is encouraging to see the performance improvement of the proposed ideas on two big datasets, it is the last part of the experiments, *i.e.*, dataset size and performance, that really intrigues us. With infinitely large datasets that cover the entirety of an arbitrarily complex problem domain, it is arguably possible that a simple over-parameterized model could solve it. However, in reality, there is barely any chance that one would observe all the domain, yet humans still learn quite efficiently how the hidden rules work. We believe this is the core where the real intelligence lies: learning from only a few samples and generalizing to the extreme. Even though CoPINet already demonstrates better learning efficiency, it would be ideal to have models capable of few-shot learning in the task of RPM. Without massive datasets, it would be a real challenge, and we hope the paper could call for future research into it.

Performance, however, is definitely not the end goal in the line of research on relational and analogical visual reasoning: other dimensions for measurements include generalization, generability, and transferability. Is it possible for a model to be trained on a single configuration and generalize to other settings? Can we generate the final answer based on the given context panels, in a similar way to the top-down and bottom-up method jointly applied by humans for reasoning? Can we transfer the relational and geometric knowledge required in the reasoning task from other tasks? Questions like these are far from being answered. While Zhang *et al.* [ZGJ19] show in the experiments that neural models do possess a certain degree of generalizability, the testing accuracy is far from satisfactory. In the meantime, there are a plethora of discriminative approaches towards solving reasoning problems in question answering, but generative methods and combined methods are lacking. The relational and analogical reasoning was initially introduced as a way to measure a human's intelligence, without training humans on the task. However, current settings uniformly reformulate it as a learning problem rather than a transfer problem, contradictory to why the task was started. Up to now, there has been barely any work that measures how knowledge on another task could be transferred to this one. We believe that significant advances in these dimensions would possibly enable Artificial Intelligence (AI) models to go beyond data fitting and acquire symbolized knowledge.

While modern computer vision techniques to solve Raven's Progressive Matrices (RPM) are based on neural networks, a promising ingredient is nowhere to be found: Gestalt psychology. Traces of the perceptual grouping and figure-ground organization are gradually faded out in the most recent wave of deep learning. However, the principles of grouping, both classical (*e.g.*, proximity, closure, and similarity) and new (*e.g.*, synchrony, element, and uniform connectedness) play an essential role in RPM, as humans arguably solve these problems by first figuring out groups and then applying the rules. We anticipate that modern deep learning methods integrated with the tradition of conceptual and theoretical foundations of the Gestalt approach would further improve models on concept induction tasks.

CHAPTER 4

Few-shot Concept Abduction via Probabilistic Abduction and Execution

Spatial-temporal reasoning is a challenging task in Artificial Intelligence (AI) due to its demanding but unique nature: a theoretic requirement on *representing* and *reasoning* based on spatial-temporal knowledge in mind, and an applied requirement on a high-level cognitive system capable of *navigating* and *acting* in space and time. Recent works have focused on an abstract reasoning task of this kind—Raven's Progressive Matrices (RPM). Despite the encouraging progress on RPM that achieves human-level performance in terms of accuracy, modern approaches have neither a treatment of human-like reasoning on generalization, nor a potential to generate answers. To fill in this gap, we propose a neuro-symbolic **Probabilistic Abduction and Execution (PrAE)** learner; central to the PrAE learner is the process of probabilistic abduction and execution on a probabilistic scene representation, akin to the mental manipulation of objects. Specifically, we disentangle perception and reasoning from a monolithic model. The neural visual perception frontend predicts objects' attributes, later aggregated by a scene inference engine to produce a probabilistic scene representation. In the symbolic logical reasoning backend, the PrAE learner uses the representation to **abduce** the hidden rules. An answer is predicted by **executing** the rules on the probabilistic representation. The entire system is trained end-to-end in an analysis-by-synthesis manner **without** any visual attribute annotations. Extensive experiments demonstrate that the PrAE learner improves cross-configuration generalization and is capable of rendering an answer, in contrast to prior works that merely make a categorical choice from candidates.

4.1 Introduction

While "thinking in pictures" [Gra06], *i.e.*, spatial-temporal reasoning, is effortless and instantaneous for humans, this significant ability has proven to be particularly challenging for current machine vision systems [JSY17]. With the promising results [Gra06] that show the very ability is strongly correlated with one's logical induction performance and a crucial factor for the intellectual history of technology development, recent computational studies on the problem focus on an abstract reasoning task relying heavily on "thinking in pictures"—Raven's Progressive Matrices (RPM) [CJS90, Hun74, Rav36, RC98]. In this task, a subject is asked to pick a correct answer that best fits an incomplete figure matrix to satisfy the hidden governing rules. The ability to solve RPM-like problems is believed to be critical for generating and conceptualizing solutions to multi-step problems, which requires mental manipulation of given images over a time-ordered sequence of spatial transformations. Such a task is also believed to be characteristic of relational and analogical reasoning and an indicator of one's *fluid intelligence* [EKM84, Hof95, JBJ08, Spe27].

State-of-the-art algorithms incorporating a contrasting mechanism and perceptual inference [HSB19, ZJG19] have achieved decent performance in terms of accuracy. Nevertheless, along with the improved accuracy from deep models come critiques on its transparency, interpretability, generalization, and difficulty to incorporate knowledge. Without explicitly distinguishing perception and reasoning, existing methods use a *monolithic* model to learn correlation, sacrificing transparency and interpretability in exchange for improved performance [HSB19, HML21, SHB18, WJL20, ZGJ19, ZJG19, ZZW19]. Furthermore, as shown in experiments, deep models nearly always overfit to the training regime and cannot properly generalize. Such a finding is consistent with Fodor [FP88] and Marcus's [Mar98, Mar18] hypothesis that human-level systematic generalizability is hardly compatible with classic neural networks; Marcus postulates that a neuro-symbolic architecture should be recruited for human-level generalization [EKS18, EMQ20, EQZ19, MD19, MD20, XMY21].

Another defect of prior methods is the lack of top-down and bottom-up reasoning [ZJG19]: Human reasoning applies a *generative* process to abduce rules and execute them to synthesize a possible solution in mind, and *discriminatively* selects the most similar answer from choices [HM12]. This bi-directional reasoning is in stark contrast to discriminative-only models, solely capable of making a categorical choice.

Psychologists also call for weak attribute supervision in RPM. As isolated Amazonians, absent of schooling on primitive attributes, could still correctly solve RPM [DIP06, IPS11], an ideal computational counterpart should be able to learn it *absent of visual attribute annotations*. This weakly-supervised setting introduces unique challenges: How to jointly learn these visual attributes given only ground-truth images? With uncertainties in perception, how to abduce hidden logic relations from it? How about executing the symbolic logic on inaccurate perception to derive answers?

To support cross-configuration generalization and answer generation, we move a step further towards a neuro-symbolic model with explicit logical reasoning and human-like generative problem-solving while addressing the challenges. Specifically, we propose the *Probabilistic Abduction and Execution (PrAE)* learner; central to it is the process of abduction and execution on the probabilistic scene representation. Inspired by Fodor, Marcus, and neuro-symbolic reasoning [HMG19, MGK19, YGL20, YWG18], the PrAE learner disentangles the previous monolithic process into two separate modules: a neural visual perception frontend and a symbolic logical reasoning backend. The neural visual frontend operates on object-based representation [HMG19, KSM17, MGK19, YGL20, YWG18] and predicts conditional probability distributions on its attributes. A scene inference engine then aggregates all object attribute distributions to produce a probabilistic scene representation for the backend. The symbolic logical backend abduces, from the representation, hidden rules that govern the time-ordered sequence via inverse dynamics. An execution engine executes the rules to *generate* an answer representation in a probabilistic planning manner [GNT04, HXZ19, KKL15], instead of directly making a categorical choice among the candidates. The final choice is

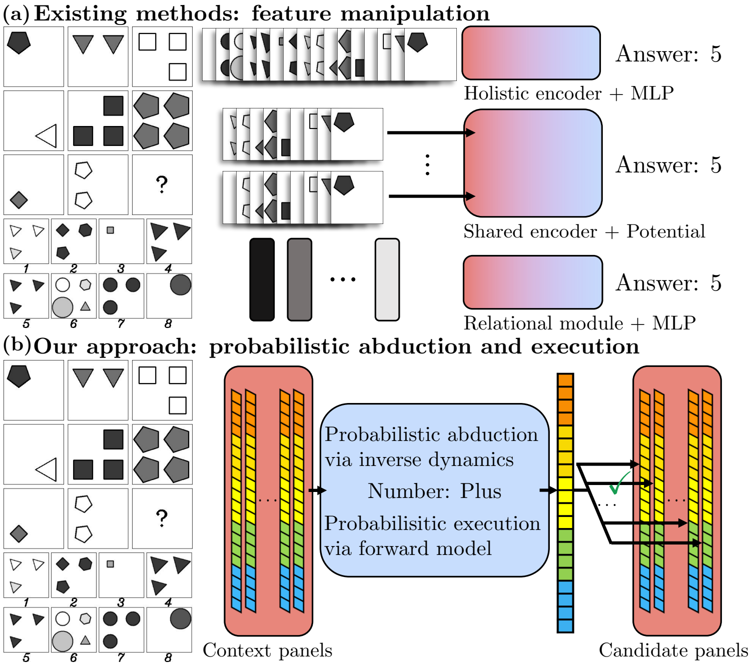

(a) Existing methods: feature manipulation
Answer: 5
Holistic encoder + MLP
Answer: 5
Shared encoder + Potential
Answer: 5
Relational module + MLP
(b) Our approach: probabilistic abduction and execution
Probabilistic abduction via inverse dynamics
Number: Plus
Probabilisitic execution via forward model
Context panels
Candidate panels
?
1
2
3
4
5
6
7
8

Figure 4.1: Differences between (a) prior methods and (b) the proposed approach. Prior methods do not explicitly distinguish perception and reasoning; instead, they use a monolithic model and only differ in how features are manipulated, lacking semantics and probabilistic interpretability. In contrast, the proposed approach disentangles this monolithic process: It perceives each panel of RPM as a set of probability distributions of attributes, performs logical reasoning to abduce the hidden rules that govern the time-ordered sequence, and executes the abduced rules to *generate* answer representations. A final choice is made based on the divergence between predicted answer distributions and each candidate's distributions; see Sec. 4.2 for a detailed comparison.

selected based on the divergence between the generated prediction and the given candidates. The entire system is trained end-to-end with a cross-entropy loss and a curricular auxiliary loss [SHB18, ZGJ19, ZJG19] *without* any visual attribute annotations. Fig. 4.1 compares the proposed PrAE learner with prior methods.

The unique design in PrAE connects perception and reasoning and offers several advantages: (i) With an intermediate probabilistic scene representation, the neural visual perception frontend and the symbolic logical reasoning backend can be *swapped* for different task domains, enabling a greater extent of module reuse and combinatorial *generalization*. (ii) Instead of blending perception and reasoning into one monolithic model without any explicit reasoning, probabilistic abduction offers a more *interpretable* account for reasoning on a logical representation. It also affords a more detailed analysis into both perception and reasoning. (iii) Probabilistic execution permits a *generative* process to be integrated into the system. Symbolic logical constraints can be transformed by the execution engine into a forward model [JR92] and applied in a probabilistic manner to predict the final scene representation, such that the entire system can be trained by analysis-by-synthesis [CHY19, Gre76, HNF19, HQX18, HQZ18, LB14, WTK17, WWX17, XLZ16, XZW19, YK06, ZWM98]. (iv) Instead of making a deterministic decision or drawing limited samples, maintaining probabilistic distributions brings in extra robustness and fault tolerance and allows gradients to be easily propagated.

This paper makes three major contributions: (i) We propose the *Probabilistic Abduction and Execution (PrAE)* learner. Unlike previous methods, the PrAE learner disentangles perception and reasoning from a monolithic model with the reasoning process realized by abduction and execution on a probabilistic scene representation. The abduction process performs interpretable reasoning on perception results. The execution process adds to the learner a generative flavor, such that the system can be trained in an analysis-by-synthesis manner without any visual attribute annotations. (ii) Our experiments demonstrate the PrAE learner achieves better generalization results compared to existing methods in the cross-configuration generalization task of RPM. We also show that the PrAE learner is capable of generating answers for RPM questions via a renderer. (iii) We present analyses into the inner functioning of both perception and reasoning, providing an interpretable account of PrAE.

4.2 Related Work

Neuro-Symbolic Visual Reasoning Neuro-symbolic methods have shown promising potential in tasks involving an interplay between vision and language and vision and causality. Qi *et al*. [QJH20, QJZ18] showed that action recognition could be significantly improved with the help of grammar parsing, and Li *et al*. [LHH20] integrated perception, parsing, and logics into a unified framework. Of particular relevance, Yi *et al*. [YWG18] first demonstrated a prototype of a neuro-symbolic system to solve VQA [AAL15], where the vision system and the language parsing system were separately trained with a final symbolic logic system applying the parsed program to deliver an answer. Mao *et al*. [MGK19] improved such a system by making the symbolic component continuous and end-to-end trainable, despite sacrificing the semantics and interpretability of logics. Han *et al*. [HMG19] built on [MGK19] and studied the metaconcept problem by learning concept embeddings. A recent work investigated temporal and causal relations in collision events [YGL20] and solved it in a way similar to [YWG18]. The proposed PrAE learner is similar to but has fundamental differences from existing neuro-symbolic methods. Unlike the method proposed by Yi *et al*. [YGL20, YWG18], our approach is end-to-end trainable and does not require intermediate visual annotations, such as ground-truth attributes. Compared to [MGK19], our approach preserves logic semantics and interpretability by explicit logical reasoning involving probabilistic abduction and execution in a probabilistic planning manner [GNT04, HXZ19, KKL15].

Computational Approaches to RPM Initially proposed as an intelligence quotient test into general intelligence and fluid intelligence [Rav36, RC98], Raven's Progressive Matrices (RPM) has received notable attention from the research community of cognitive science. Psychologists have proposed reasoning systems based on symbolic representations and discrete logics [CJS90, LF17, LFU10, LTF09]. However, such logical systems cannot handle visual uncertainty arising from imperfect perception. Similar issues also pose challenges to methods based on image similarity [LLG12, MG14, MKG14, MSD18, SG18a]. Recent works approach this problem in a data-driven manner. The first automatic RPM generation method was proposed by Wang and Su [WS15]. Santoro *et al.* [SHB18] extended it using procedural generation and introduced the WReN to solve the problem. Zhang *et al.* [ZGJ19] and Hu *et al.* [HML21] used stochastic image grammar [ZM07] and provided structural annotations to the dataset. Unanimously, existing methods do not explicitly distinguish perception and reasoning; instead, they use one monolithic neural model, sacrificing interpretability in exchange for better performance. The differences in previous methods lie in how features are manipulated: Santoro *et al.* [SHB18] used the relational module to extract final features, Zhang *et al.* [ZGJ19] stacked all panels into the channel dimension and fed them into a residual network, Hill *et al.* [HSB19] prepared the data in a contrasting manner, Zhang *et al.* [ZJG19] composed the context with each candidate and compared their potentials, Wang *et al.* [WJL20] modeled the features by a multiplex graph, and Hu *et al.* [HML21] integrated hierarchical features. Zheng *et al.* [ZZW19] studied a teacher-student setting in RPM, while Steenbrugge *et al.* [SLV18] focused on a generative approach to improve learning. Concurrent to our work, Spratley *et al.* [SEM20] unsupervisedly extracted object embeddings and conducted reasoning via a ResNet. In contrast, PrAE is designed to address cross-configuration generalization and disentangles perception and reasoning from a monolithic model, with symbolic logical reasoning implemented as probabilistic abduction and execution.

4.3 The PrAE Learner

Problem Setup In this section, we explain our approach to tackling the RPM problem. Each RPM instance consists of 16 panels: 8 context panels form an incomplete 3×3 matrix with a 9th missing entry, and 8 candidate panels for one to choose. The goal is to pick one candidate that best completes the matrix to satisfy the latent governing rules. Existing datasets [HML21, SHB18, WS15, ZGJ19] assume fixed sets of object attributes, panel

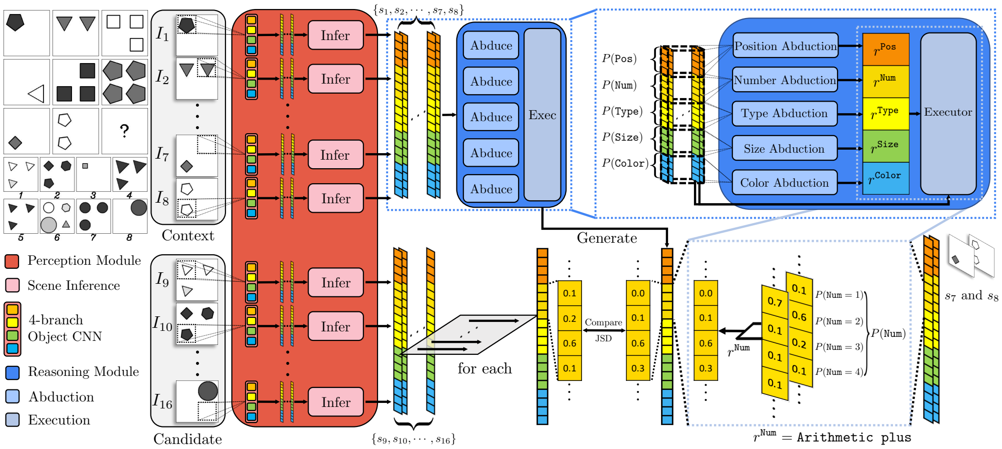


Figure 4.2: An overview of learning and reasoning of the proposed PrAE learner. Given an RPM instance, the neural perception frontend (in red) extracts probabilistic scene representation for each of the 16 panels (8 contexts + 8 candidates). The *Object CNN* sub-module takes in each image region returned by a sliding window to produce object attribute distributions (over objectiveness, type, size, and color). The *Scene Inference Engine* sub-module (in pink) aggregates object attribute distributions from all regions to produce panel attribute distributions (over position, number, type, size, and color). Probabilistic representation for context panels is fed into the symbolic reasoning backend (in blue), which abduces hidden rule distributions for all panel attributes (upper-right figure) and executes chosen rules on corresponding context panels to generate the answer representation (lower-right figure). The answer representation is compared with each candidate representation from the perception frontend; the candidate with minimum divergence from the prediction is chosen as the final answer. The lower-right figure is an example of probabilistic execution on the panel attribute of `Number`; see Sec. 4.3.2 for the exact computation process.

attributes, and rules, with each panel attribute governed by one rule. The value of a panel attribute constrains the value of the corresponding object attribute for each object in it.

Overview The proposed neuro-symbolic PrAE learner disentangles previous monolithic visual reasoning into two modules: the neural visual perception frontend and the symbolic logical reasoning backend. The frontend uses a CNN to extract object attribute distributions, later aggregated by a scene inference engine to produce panel attribute distributions. The set of all panel attribute distributions in a panel is referred to as its *probabilistic scene representation*. The backend retrieves this compact scene representation and performs logical abduction and execution in order to predict the answer representation in a generative manner. A final choice is made based on the divergence between the prediction and each candidate. Using REINFORCE [Wil92], the entire system is trained *without attribute annotations* in a curricular manner; see Fig. 4.2 for an overview of PrAE.

4.3.1 Neural Visual Perception

The neural visual perception frontend operates on each of the 16 panels *independently* to produce probabilistic scene representation. It has two sub-modules: object CNN and scene inference engine.

Object CNN Given an image panel I, a sliding window traverses its spatial domain and feeds each image region into a 4-branch CNN. The 4 CNN branches use the same LeNet-like architecture [LBB98] and produce the probability distributions of object attributes, including objectiveness (whether the image region has an object), type, size, and color. Of note, the distributions of type, size, and color are conditioned on objectiveness being true. Attribute distributions of each image region are kept and sent to the scene inference engine to produce panel attribute distributions.

Scene Inference Engine The scene inference engine takes in the outputs of object CNN and produces panel attribute distributions (over position, number, type, size, and color) by marginalizing over the set of object attribute distributions (over objectiveness, type, size, and color). Take the panel attribute of `Number` as an example: Given N objectiveness probability distributions produced by the object CNN for N image regions, the probability of a panel having k objects can be computed as

$$P(\texttt{Number} = k) = \sum_{\substack{B^o \in \{0,1\}^N \\ |B^o| = k}} \prod_{j=1}^{N} P(b^o_j = B^o_j), \tag{4.1}$$

where B^o is an ordered binary sequence corresponding to objectiveness of the N regions, $|\cdot|$ the number of 1 in the sequence, and $P(b^o_j)$ the objectiveness distribution of the jth region. We assume $k \geqslant 1$ in each RPM panel, leave $P(\texttt{Number} = 0)$ out, and renormalize the probability to have a sum of 1. The panel attribute distributions for position, type, size, and color, can be computed similarly.

We refer to the set of all panel attribute distributions in a panel its *probabilistic scene representation*, denoted as s, with the distribution of panel attribute a denoted as $P(s^a)$.

4.3.2 Symbolic Logical Reasoning

The symbolic logical reasoning backend collects probabilistic scene representation from 8 context panels, abduces the probability distributions over hidden rules on each panel attribute, and executes them on corresponding panels of the context. Based on a prior study [CJS90], we assume a set of symbolic logical constraints describing rules is available. For example, the `Arithmetic plus` rule on `Number` can be represented as: for each row (column), $\forall l, m \geqslant 1$

$$(\texttt{Number}_1 = m) \wedge (\texttt{Number}_2 = l) \wedge (\texttt{Number}_3 = m + l), \tag{4.2}$$

where $\texttt{Number}_i$ denotes the number of objects in the ith panel in a row (column). With access to such constraints, we use inverse dynamics to abduce the rules in an instance. They can also be transformed into a forward model and executed on discrete symbols: For instance, `Arithmetic plus` deterministically adds `Number` in the first two panels to obtain the `Number` of the last panel.

Probabilistic Abduction Given the probabilistic scene representation of 8 context panels, the probabilistic abduction engine calculates the probability of rules for each panel attribute via inverse dynamics. Formally, for each rule r on a panel attribute a,

$$P(r^a \mid I_1, \dots, I_8) = P(r^a \mid I_1^a, \dots, I_8^a), \tag{4.3}$$

where I_i denotes the ith context panel, and I_i^a the component of context panel I_i corresponding to a. Note Eq. (4.3) generalizes inverse dynamics [JR92] to 8 states, in contrast to that of a conventional MDP.

To model $P(r^a \mid I_1^a, \dots, I_8^a)$, we leverage the compact probabilistic scene representation with respect to attribute a and logical constraints:

$$P(r^a \mid I_1^a, \dots, I_8^a) \propto \sum_{S^a \in \texttt{valid}(r^a)} \prod_{i=1}^{8} P(s_i^a = S_i^a), \tag{4.4}$$

where $\texttt{valid}(\cdot)$ returns a set of attribute value assignments of the context panels that satisfy the logical constraints of r^a, and i indexes into context panels. By going over all panel attributes, we have the distribution of hidden rules for each of them.

Take `Arithmetic plus` on `Number` as an example. A row-major assignment for context panels can be $[1, 2, 3, 1, 3, 4, 1, 2]$ (as in Fig. 4.2), whose probability is computed as the product of each panel having k objects as in Eq. (4.1). Summing it with other assignment probabilities gives an unnormalized rule probability.

We note that the set of valid states for each r^a is a product space of valid states on each row (column). Therefore, we can perform partial marginalization on each row (column) first and aggregate them later to avoid directly marginalizing over the entire space. This decomposition will help reduce computation and mitigate numerical instability.

Probabilistic Execution For each panel attribute a, the probabilistic execution engine chooses a rule from the abduced rule distribution and executes it on corresponding context panels to predict, in a generative fashion, the panel attribute distribution of an answer. While traditionally, a logical forward model only works on discrete symbols, we follow a generalized notion of probabilistic execution as done in probabilistic planning [HXZ19, KKL15]. The probabilistic execution could be treated as a distribution transformation that redistributes the probability mass based on logical rules. For a binary rule r on a,

$$P(s_3^a = S_3^a) \propto \sum_{\substack{(S_2^a, S_1^a) \in \texttt{pre}(r^a) \\ S_3^a = f(S_2^a, S_1^a; r^a)}} P(s_2^a = S_2^a) P(s_1^a = S_1^a), \tag{4.5}$$

where f is the forward model transformed from logical constraints and $\texttt{pre}(\cdot)$ the rule precondition set. Predicted distributions of panel attributes compose the final probabilistic scene representation s_f.

As an example of `Arithmetic plus` on `Number`, 4 objects result from the addition of $(1, 3)$, $(2, 2)$, and $(3, 1)$. The probability of an answer having 4 objects is the sum of the instances' probabilities.

During training, the execution engine samples a rule from the abduced probability. During testing, the most probable rule is chosen.

Candidate Selection With a set of predicted panel attribute distributions, we compare it with that from each candidate answer. We use the Jensen–Shannon Divergence (JSD) [Lin91] to quantify the divergence between the prediction and the candidate, *i.e.*,

$$d(s_f, s_i) = \sum_a \mathbb{D}_{\text{JSD}}(P(s_f^a) \mid\mid P(s_i^a)), \tag{4.6}$$

where the summation is over panel attributes and i indexes into the candidate panels. The candidate with minimum divergence will be chosen as the final answer.

Discussion The design of reasoning as probabilistic abduction and execution is a computational and interpretable counterpart to human-like reasoning in RPM [CJS90]. By abduction, one infers the hidden rules from context panels. By executing the abduced rules, one obtains a probabilistic answer representation. Such a probabilistic representation is compared with all candidates available; the most similar one in terms of divergence is picked as the final answer. Note that the probabilistic execution adds the generative flavor into reasoning: Eq. (4.5) depicts the predicted panel attribute distribution, which can be sampled and sent to a rendering engine for panel generation. The entire process resembles bi-directional inference and combines both top-down and bottom-up reasoning missing in prior works. In the meantime, the design addresses challenges mentioned in Sec. 4.1 by marginalizing over perception and abducing and executing rules probabilistically.

4.3.3 Learning Objective

During training, we transform the divergence in Eq. (4.6) into a probability distribution by

$$P(\text{Answer} = i) \propto \exp(-d(s_f, s_i)) \tag{4.7}$$

and minimize the cross-entropy loss. Note that the learning procedure follows a general paradigm of analysis-by-synthesis [CHY19, Gre76, HNF19, HQX18, HQZ18, LB14, WTK17, WWX17, XLZ16, XZW19, YK06, ZWM98]: The learner synthesizes a result and measures difference analytically.

As the reasoning process involves rule selection, we use REINFORCE [Wil92] to optimize:

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)], \tag{4.8}$$

where θ denotes the trainable parameters in the object CNN, $P(r)$ packs the rule distributions over all panel attributes, ℓ is the cross-entropy loss, and y is the ground-truth answer. Note that here we make explicit the dependency of the answer distribution on rules, as the predicted probabilistic scene representation s_f is dependent on the rules chosen.

In practice, the PrAE learner experiences difficulty in convergence with cross-entropy loss only, as the object CNN fails to produce meaningful object attribute predictions at the early stage of training. To resolve this issue, we jointly train the PrAE learner to optimize the auxiliary loss, as discussed in recent literature [SHB18, ZGJ19, ZJG19]. The auxiliary loss regularizes the perception module such that the learner produces the correct rule prediction.

The final objective is

$$\min_{\theta} \mathbb{E}_{P(r)}[\ell(P(\text{Answer}; r), y)] + \sum_{a} \lambda^a \ell(P(r^a), y^a), \tag{4.9}$$

where λ^a is the weight coefficient, $P(r^a)$ the distribution of the abduced rule on a, and y^a the ground-truth rule. In reinforcement learning terminology, one can treat the cross-entropy loss as the negative reward and the auxiliary loss as behavior cloning [SB98].

4.3.4 Curriculum Learning

In preliminary experiments, we notice that accurate objectiveness prediction at the early stage is essential to the success of the learner, while learning without auxiliary will reinforce the perception system to produce more accurate object attribute predictions in the later stage when all branches of the object CNN are already warm-started. This observation is consistent with human learning: One learns object attributes only after they can correctly distinguish objects from the scene, and their perception will be enhanced with positive signals from the task.

Based on this observation, we train our PrAE learner in a 3-stage curriculum [BLC09]. In the first stage, only parameters corresponding to objectiveness are trained. In the second stage, objectiveness parameters are frozen while weights responsible for type, size, and color prediction are learned. In the third stage, we perform joint fine-tuning for the entire model via REINFORCE [Wil92].

4.4 Experiments

We demonstrate the efficacy of the proposed PrAE learner in RPM. In particular, we show that the PrAE learner achieves the best performance among all baselines in the cross-configuration generalization task of RPM. In addition, the modularized perception and reasoning process allows us to probe into how each module performs in the RPM task and

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
WReN	9.86/14.87	8.65/14.25	29.60/20.50	9.75/15.70	4.40/13.75	5.00/13.50	5.70/14.15	5.90/12.25
LSTM	12.81/12.52	12.70/12.55	13.80/13.50	12.90/11.35	12.40/14.30	12.10/11.35	12.45/11.55	13.30/13.05
LEN	12.29/13.60	11.85/14.85	41.40/18.20	12.95/13.35	3.95/12.55	3.95/12.75	5.55/11.15	6.35/12.35
CNN	14.78/12.69	13.80/11.30	18.25/14.60	14.55/11.95	13.35/13.00	15.40/13.30	14.35/11.80	13.75/12.85
MXGNet	20.78/13.07	12.95/13.65	37.05/13.95	24.80/12.50	17.45/12.50	16.80/12.05	18.05/12.95	18.35/13.90
ResNet	24.79/13.19	24.30/14.50	25.05/14.30	25.80/12.95	23.80/12.35	27.40/13.55	25.05/13.40	22.15/11.30
ResNet+DRT	31.56/13.26	31.65/13.20	39.55/14.30	35.55/13.25	25.65/12.15	32.05/13.10	31.40/13.70	25.05/13.15
SRAN	15.56/29.06	18.35/37.55	38.80/38.30	17.40/29.30	9.45/29.55	11.35/28.65	5.50/21.15	8.05/18.95
CoPINet	52.96/22.84	49.45/24.50	61.55/31.10	**52.15**/25.35	68.10/20.60	65.40/19.85	39.55/19.00	34.55/19.45
PrAE Learner	**65.03**/**77.02**	**76.50**/**90.45**	**78.60**/**85.35**	28.55/**45.60**	**90.05**/**96.25**	**90.85**/**97.35**	**48.05**/**63.45**	**42.60**/**60.70**
Human	84.41	95.45	81.82	79.55	86.36	81.81	86.36	81.81

Table 4.1: Model performance (%) on RAVEN / I-RAVEN. All models are trained on 2x2Grid only. Acc denotes the mean accuracy. Following Zhang *et al.* [ZGJ19], L-R is short for the Left-Right configuration, U-D Up-Down, O-IC Out-InCenter, and O-IG Out--InGrid.

Object Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Objectiveness	93.81/95.41	96.13/96.07	99.79/99.99	99.71/97.98	99.56/95.00	99.86/94.84	71.73/88.05	82.07/95.97
Type	86.29/89.24	89.89/89.33	99.95/95.93	83.49/85.96	99.92/92.90	99.85/97.84	91.55/91.86	66.68/70.85
Size	64.72/66.63	68.45/69.11	71.26/73.20	71.42/62.02	73.00/85.08	73.41/73.45	53.54/62.63	44.36/40.95
Color	75.26/79.45	75.15/75.65	85.15/87.81	62.69/69.94	85.27/83.24	84.45/81.38	84.91/75.32	78.48/82.84

Table 4.2: Accuracy (%) of the object CNN on each attribute, reported as RAVEN / I-RAVEN. The CNN module is trained with the PrAE learner on 2x2Grid only without any visual attribute annotations. Acc denotes the mean accuracy on each attribute.

analyze the PrAE learner's strengths and weaknesses. Furthermore, we show that probabilistic scene representation learned by the PrAE learner can be used to generate an answer when equipped with a rendering engine.

Panel Attribute	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
Pos/Num	90.53/91.67	-	90.55/90.05	92.80/94.10	-	-	-	88.25/90.85
Type	94.17/92.15	100.00/95.00	99.75/95.30	63.95/68.40	100.00/99.90	100.00/100.00	100.00/100.00	86.08/77.60
Size	90.06/88.33	98.95/99.00	90.45/89.90	65.30/70.45	98.15/96.78	99.45/92.45	93.08/96.13	77.35/70.78
Color	87.38/87.25	97.60/93.75	88.10/85.35	37.45/45.65	98.90/92.38	99.40/98.43	92.90/97.23	73.75/79.48

Table 4.3: Accuracy (%) of the probabilistic abduction engine on each attribute, reported as RAVEN / I-RAVEN. The PrAE learner is trained on 2x2Grid only. Acc denotes the mean accuracy on each attribute.

4.4.1 Experimental Setup

We evaluate the proposed PrAE learner on RAVEN [ZGJ19] and I-RAVEN [HML21]. Both datasets consist of 7 distinct RPM configurations, each of which contains $10,000$ samples, equally divided into 6 folds for training, 2 folds for validation, and 2 folds for testing. We compare our PrAE learner with simple baselines of LSTM, CNN, and ResNet, and strong baselines of WReN [SHB18], ResNet+DRT [ZGJ19], LEN [ZZW19], CoPINet [ZJG19], MXGNet [WJL20], and SRAN [HML21]. To measure cross-configuration generalization, we train all models using the 2x2Grid configuration due to its proper complexity for probability marginalization and a sufficient number of rules on each panel attribute. We test the models on all *other* configurations. All models are implemented in PyTorch [PGC17] and optimized using ADAM [KB14] on an Nvidia Titan Xp GPU. For numerical stability, we use log probability in PrAE.

4.4.2 Cross-Configuration Generalization

Tab. 4.1 shows the cross-configuration generalization performance of different models. While advanced models like WReN, LEN, MXGNet, and SRAN have fairly good fitting performance on the training regime, these models fail to learn transferable representation for other configurations, which suggests that they do not learn logics or any forms of abstraction but visual appearance only. Simpler baselines like LSTM, CNNs, ResNet, and ResNet+DRT show less severe overfitting, but neither do they demonstrate satisfactory performance. This effect indicates that using only deep models in abstract visual reasoning makes it very difficult to acquire the generalization capability required in situations with similar inner mechanisms but distinctive appearances. By leveraging the notion of contrast, CoPINet improves generalization performance by a notable margin.

Equipped with symbolic reasoning and neural perception, not only does the PrAE learner achieve the best performance among all models, but it also shows performance better than humans on three configurations. Compared to baselines trained on the full dataset, the PrAE learner surpasses all other models on the 2x2Grid domain, despite other models seeing 6 times more data. The PrAE learner does not exhibit strong overfitting either, achieving comparable and sometimes better performance on Center, L-R, and U-D. However, limitations of the PrAE learner do exist. In cases with overlap (O-IC and O-IG), the performance decreases, and a devastating result is observed on 3x3Grid. The first failure is due to the domain shift in the region appearance that neural models cannot handle, and the second could be attributed to marginalization over probability distributions of multiple objects in 3x3Grid, where uncertainties from all objects accumulate, leading to inaccurate abduced rule distributions. These observations are echoed in our analysis shown next.

4.4.3 Analysis on Perception and Reasoning

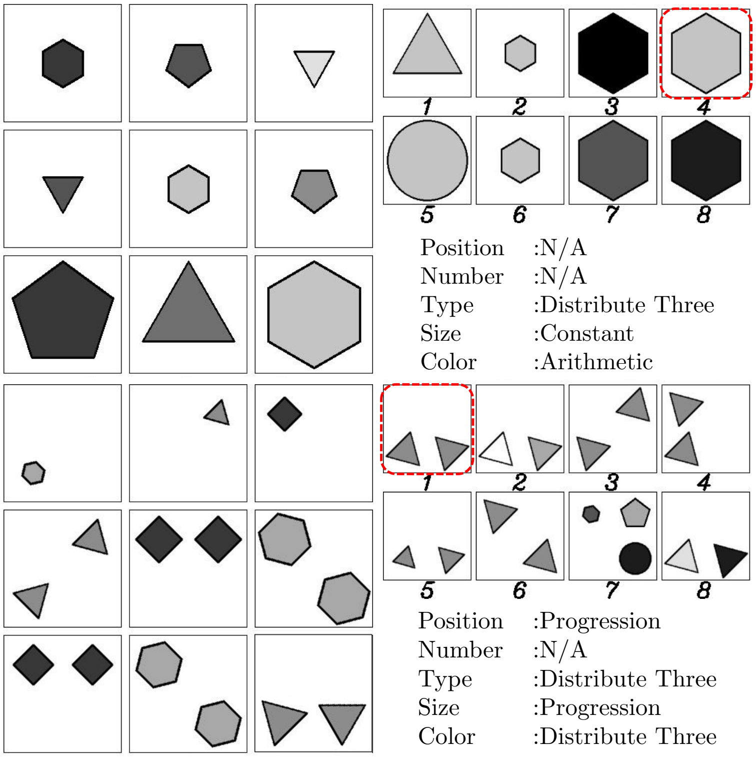
RAVEN and I-RAVEN provide multiple levels of annotations for us to analyze our modularized PrAE learner. Specifically, we use the region-based attribute annotations to evaluate our object CNN in perception. Note that the object CNN is not trained using any attribute annotations. We also use the ground-truth rule annotations to evaluate the accuracy of the probabilistic abduction engine.

Tab. 4.2 details the analysis of perception using the object CNN: It achieves reasonable performance on object attribute prediction, though not trained with any visual attribute annotations. The model shows a relatively accurate prediction of objectiveness in order to solve an RPM instance. Compared to the size prediction accuracy, the object CNN is better at predicting texture-related attributes of type and color. The object CNN has similar results on 2x2Grid, L-R, and U-D. However, referencing Tab. 4.1, we notice that 2x2Grid requires marginalization over more objects, resulting in an inferior performance. Accuracy further drops on configurations with overlap, leading to unsatisfactory results on O-IC and O-IG. For 3x3Grid, more accurate predictions are necessary as uncertainties accumulate from probabilities over multiple objects.

Tab. 4.3 details the analysis on reasoning, showing how the probabilistic abduction engine performs on rule prediction for each attribute across different configurations. Since rules on position and number are exclusive, we merge their performance as Pos/Num. As Center, L-R, U-D, and O-IC do not involve rules on Pos/Num, we do not measure the abduction performance on them. We note that, in general, the abduction engine shows good performance on all panel attributes, with a perfect prediction on type in certain configurations. However, the design of abduction as probability marginalization is a double-edged sword. While the object CNN's performance on size prediction is only marginally different on 2x2Grid and 3x3Grid in RAVEN, their abduction accuracies drastically vary. The difference occurs because uncertainties on object attributes accumulate during marginalization as the number of objects increases, eventually leading to poor performance on rule prediction and answer selection. However, on configurations with fewer objects, unsatisfactory object attribute predictions can still produce accurate rule predictions. Note there is no guarantee that a correct rule will necessarily lead to a correct final choice, as the selected rule still operates on panel attribute distributions inferred from object attribute distributions.

4.4.4 Generation Ability

One unique property of the proposed PrAE learner is its ability to directly generate a panel from the predicted representation when a rendering engine is given. The ability resembles the bi-directional top-down and bottom-up reasoning, adding a generative flavor commonly



Position :N/A
Number :N/A
Type :Distribute Three
Size :Constant
Color :Arithmetic

Position :Progression
Number :N/A
Type :Distribute Three
Size :Progression
Color :Distribute Three

Figure 4.3: Two RPM instances with the final 9th panels filled by our generation results. The ground-truth selections are highlighted in red squares, and the ground-truth rules in each instance are listed. There are no rules on position and number in the first instance of the Center configuration, and the rules on position and number are exclusive in the second instance of 2x2Grid.

ignored in prior discriminative-only approaches [HSB19, HML21, SHB18, WJL20, ZGJ19, ZJG19, ZZW19]. As the PrAE learner predicts final panel attribute distributions and is trained in an analysis-by-synthesis manner, we can sample panel attribute values from the predicted distributions and render the final answer using a rendering engine. Here, we use the rendering program released with RAVEN [ZGJ19] to show the generation ability of the PrAE learner. Fig. 4.3 shows examples of the generation results. Note that one of our generations is slightly different from the ground-truth answer due to random sampling of rotations during rendering. However, it still follows the rules in the problem and should be considered as a correct answer.

4.5 Conclusion and Discussion

We propose the *Probabilistic Abduction and Execution (PrAE)* learner for spatial-temporal reasoning in Raven's Progressive Matrices (RPM) that decomposes the problem-solving process into neural perception and logical reasoning. While existing methods on RPM are merely discriminative, the proposed PrAE learner is a hybrid of generative models and discriminative models, closing the loop in a human-like, top-down bottom-up bi-directional reasoning process. In the experiments, we show that the PrAE learner achieves the best performance on the cross-configuration generalization task on RAVEN and I-RAVEN. The modularized design of the PrAE learner also permits us to probe into how perception and reasoning work independently during problem-solving. Finally, we show the unique generative property of the PrAE learner by filling in the missing panel with an image produced by the values sampled from the probabilistic scene representation.

However, the proposed PrAE learner also has limits. As shown in our experiments, probabilistic abduction can be a double-edged sword in the sense that when the number of objects increases, uncertainties over multiple objects will accumulate, making the entire process sensitive to perception performance. Also, complete probability marginalization introduces a challenge for computational scalability; it prevents us from training the PrAE learner on more complex configurations such as 3x3Grid. One possible solution might be a discrete abduction process. However, jointly learning such a system is non-trivial. It is also difficult for the learner to perceive and reason based on lower-level primitives, such as lines and corners. While, in theory, a generic detector of lines and corners should be able to resolve this issue, no well-performing systems exist in practice, except those with strict handcrafted detection rules, which would miss the critical probabilistic interpretations in the entire framework. The PrAE learner also requires strong prior knowledge about the underlying logical relations to work, while an ideal method should be able to induce the hidden rules by itself. Though a precise induction mechanism is still unknown for humans, an emerging computational technique of bi-level optimization [FAL17, ZZZ19] may be able to house perception and induction together into a general optimization framework.

While we answer questions about generalization and generation in RPM, one crucial question remains to be addressed: How perception learned from other domains can be transferred and used to solve this abstract reasoning task. Unlike humans that arguably apply knowledge learned from elsewhere to solve RPM, current systems still need training on the same task to acquire the capability. While feature transfer is still challenging for computer vision, we anticipate that progress in answering transferability in RPM will help address similar questions [ZJE21, ZZZ20, ZGF20] and further advance the field.

CHAPTER 5

Learning Algebraic Representation for Few-shot Concept Induction

Is intelligence realized by connectionist or classicist? While connectionist approaches have achieved superhuman performance, there has been growing evidence that such task-specific superiority is particularly fragile in *systematic generalization*. This observation lies in the central debate between connectionist and classicist, wherein the latter continually advocates an *algebraic* treatment in cognitive architectures. In this work, we follow the classicist's call and propose a hybrid approach to improve systematic generalization in reasoning. Specifically, we showcase a prototype with algebraic representation for the abstract spatial-temporal reasoning task of RPM and present the ALANS learner. The ALANS learner is motivated by abstract algebra and the representation theory. It consists of a neural visual perception frontend and an algebraic abstract reasoning backend: the frontend summarizes the visual information from object-based representation, while the backend transforms it into an algebraic structure and induces the hidden operator on the fly. The induced operator is later executed to predict the answer's representation, and the choice most similar to the prediction is selected as the solution. Extensive experiments show that by incorporating an algebraic treatment, the ALANS learner outperforms various pure connectionist models in domains requiring systematic generalization. We further show the generative nature of the learned algebraic representation; it can be decoded by isomorphism to generate an answer.

5.1 Introduction

"Thought is in fact a kind of Algebra."

—William James [Jam91]

Imagine you are given two alphabetical sequences of "c, b, a" and "d, c, b", and asked to fill in the missing element in "$e, d, ?$". In nearly no time will one realize the answer to be c. However, more surprising for human learning is that, effortlessly and instantaneously, we can "freely generalize" [Mar01] the solution to any partial consecutive ordered sequences. While believed to be innate in early development for human infants [MVR99], such systematic generalizability has constantly been missing and proven to be particularly challenging in existing connectionist models [BMN19, LB18]. In fact, such an ability to entertain a given thought and semantically related contents strongly implies an abstract algebra-like treatment [FP88]; in literature, it is referred to as the "language of thought" [Fod75], "physical symbol system" [New80], and "algebraic mind" [Mar01]. However, in stark contrast, existing connectionist models tend only to capture statistical correlation [Cho19, KSM17, LB18], rather than providing any account for a structural inductive bias where systematic algebra can be carried out to facilitate generalization.

This contrast instinctively raises a question—what constitutes such an *algebraic* inductive bias? We argue that the foundation of modeling counterpart to the algebraic treatment in early human development [Mar01, MVR99] lies in algebraic computations set up on mathematical axioms, a form of formalized human intuition and the beginning of modern mathematical reasoning [Hea56, Mad88]. Of particular importance to the algebra's basic building blocks is the Peano Axiom [Pea89]. In the Peano Axiom, the essential components of algebra, the algebraic set, and corresponding operators over it, are governed by three statements: (1) the existence of at least one element in the field to study ("zero" element), (2) a successor function that is recursively applied to all elements and can, therefore, span the entire field, and (3) the principle of mathematical induction. Building on such a fundamental axiom, we begin to form the notion of an algebraic set and induce the operator to construct an algebraic structure. We hypothesize that such an algebraic treatment set up on fundamental axioms is essential for a model's systematic generalizability, the lack of which will only make it sub-optimal.

To demonstrate the benefits of adopting such an algebraic treatment in systematic generalization, we showcase a prototype for Raven's Progressive Matrices (RPM) [Rav36, RC98], an exemplar task for abstract spatial-temporal reasoning [SHB18, ZGJ19]. In this task, an agent is given an incomplete 3 × 3 matrix consisting of eight context panels with the last one missing, and asked to pick one answer from a set of eight choices that best completes the matrix. Human's reasoning capability of solving this abstract reasoning task has been commonly regarded as an indicator of "general intelligence" [CJS90] and "fluid intelligence" [Hof95, JBJ08, Spe23, Spe27]. In spite of the task being one that ideally requires abstraction, algebraization, induction, and generalization [CJS90, Rav36, RC98], recent endeavors unanimously propose pure connectionist models that attempt to circumvent such intrinsic cognitive requirements [HML20, SHB18, WJL20, WDG20, ZGJ19, ZJG19, ZZW19]. However, these methods' inefficiency is also evident in systematic generalization; they struggle to extrapolate to domains beyond training [SHB18, ZJG19], shown also in this paper.

To address the issue, we introduce an ALgebra-Aware Neuro-Semi-Symbolic (ALANS) learner. At a high-level, the ALANS learner is embedded in a general neuro-symbolic architecture [HMG19, MGK19, YGL20, YWG18] but has the on-the-fly operator learnability, hence **semi-symbolic**. Specifically, it consists of a neural visual perception frontend and an algebraic abstract reasoning backend. For each RPM instance, the neural visual perception frontend first slides a window over each panel to obtain the object-based representation [KSM17, WTK17] for every object. A belief inference engine latter aggregates all object-based representation in each panel to produce the probabilistic *belief state*. The algebraic abstract reasoning backend then takes the belief states of the eight context panels, treats them as snapshots on an algebraic structure, lifts them into a matrix-based algebraic representation built on the Peano Axiom and the representation theory [Hum12], and induces the hidden operator in the algebraic structure by solving an inner optimization problem [Bar13, CMS07, ZZZ19]. The answer's algebraic representation is predicted by executing the induced operator: its corresponding set element is decoded by isomorphism, and the final answer is selected as the one most similar to the prediction.

The ALANS learner enjoys several benefits in abstract reasoning with an algebraic treatment:

1. Unlike previous monolithic models, the ALANS learner offers a more **interpretable** account of the entire abstract reasoning process: the neural visual perception frontend extracts object-based representation and produces belief states of panels by explicit probability inference, whereas the algebraic abstract reasoning backend induces the hidden operator in the algebraic structure. The final answer's representation is obtained by executing the induced operator, and the choice panel with minimum distance is selected. This process much resembles the top-down bottom-up strategy in human reasoning missed in recent literature [HML20, SHB18, WJL20, WDG20, ZGJ19, ZJG19, ZZW19]: humans reason by inducing the hidden relation, executing it to generate a feasible solution in mind, and choosing the most similar answer available [CJS90].

2. While keeping the semantic interpretability and end-to-end trainability in existing neuro-symbolic frameworks [HMG19, MGK19, YGL20, YWG18], ALANS is **semi-symbolic** in the sense that the symbolic operator can be learned and concluded on the fly without manual definition for every one of them. Such an inductive ability also enables a greater extent of the desired generalizability.

3. By decoding the predicted representation in the algebraic structure, we can also generate an answer that satisfies the hidden relation in the context.

This work makes three major contributions. (1) We propose the ALANS learner, a neuro-semi-symbolic design, in contrast to existing monolithic models. (2) To demonstrate the efficacy of incorporating an algebraic treatment in reasoning, we show the superior systematic generalization ability of the proposed ALANS learner in various extrapolatory RPM domains. (3) We present analyses into both neural visual perception and algebraic abstract reasoning.

5.2 Related Work

5.2.1 Quest for Symbolized Manipulation

The idea to treat thinking as a mental language can be dated back to Augustine [Aug76, Wit53]. Since the 1970s, this school of thought has undergone a dramatic revival as the quest for symbolized manipulation in cognitive modeling, such as "language of thought" [Fod75], "physical symbol system" [New80], and "algebraic mind" [Mar01]. In their study, connectionist's task-specific superiority and inability to generalize beyond training [Cho19, KSM17, SHB18, ZGJ19] have been hypothetically linked to a lack of such symbolized algebraic manipulation [Cho19, LB18, Mar20]. With evidence that an algebraic treatment adopted in early human development [MVR99] can potentially address the issue [BMN19, MGK19, Mar20], classicist [FP88] approaches for generalizable reasoning used in programs [McC60] and blocks world [Win71] have resurrected. As a hybrid approach to bridge connectionist and classicist, recent developments lead to neuro-symbolic architectures. In particular, the community of theorem proving has been one of the earliest to endorse the technique [GBG12, RR17, SG16]: ∂ILP [EG18] and NLM [DML18] make inductive programming end-to-end, and DeepProbLog [MDK18] connects learning and reasoning. Recently, Hudson and Manning [HM19] propose NSM for visual question answering where a probabilistic graph is used for reasoning. Yi *et al.* [YWG18] demonstrate a neuro-symbolic prototype for the same task where a perception module and a language parsing module are separately trained, with the predefined logic operators associated with language tokens chained to process the visual information. Mao *et al.* [MGK19] soften the predefined operators to afford end-to-end training with only question answers. Han *et al.* [HMG19] use the hybrid architecture for metaconcept learning.

Yi *et al*. [YGL20] and Chen *et al*. [CMW20] show how neuro-symbolic models can handle explanatory, predictive, and counterfactual questions in temporal and causal reasoning. Lately, NeSS [CLY20] exemplifies an algorithmic stack machine that can be used to improve generalization in language learning. ALANS follows the classicist's call but adopts a neuro-*semi*-symbolic architecture: it is end-to-end trainable as opposed to Yi *et al*. [YGL20, YWG18] and the operator can be learned and concluded on the fly without manual specification.

5.2.2 Abstract Visual Reasoning

Recent works by Santoro *et al*. [SHB18] and Zhang *et al*. [ZGJ19] arouse the community's interest in abstract visual reasoning; the task of Raven's Progressive Matrices (RPM) is introduced as such a measure for intelligent agents. As an intelligence quotient test for humans [Rav36, RC98], RPM is believed to be strongly correlated with human's general intelligence [CJS90] and fluid intelligence [Hof95, JBJ08, Spe23, Spe27]. Early RPM-solving systems employ symbolic representation based on hand-designed features and assume access to the underlying logics [CJS90, LF17, LFU10, LTF09]. Another stream of research on RPM recruits similarity-based metrics to select the most similar answer from the choices [HIL22, LLG12, MG14, MKG14, MSD18, SG18a]. However, these visual or semantic features are unable to handle uncertainty from imperfect perception, and directly assuming access to the logic operations simplifies the problem. Recently proposed data-driven approaches arise from the availability of large datasets: Santoro *et al*. [SHB18] extend a pedagogical RPM generation method [WS15], whereas Zhang *et al*. [ZGJ19] use a stochastic image grammar [ZM07] and introduce structural annotations in it, which Hu *et al*. [HML20] further refine to avoid shortcut solutions by statistics in candidate panels. Despite the fact that RPM intrinsically requires one to perform abstraction, algebraization, induction, and generalization, existing methods bypass such cognitive requirements using a single feedforward pass in connectionist models: Santoro *et al*. [SHB18] use a relational module [SRB17], Steenbrugge *et al*. [SLV18] augment it with a VAE [KW13], Zhang *et* *al.* [ZGJ19] assemble a dynamic tree, Hill *et al.* [HSB19] arrange the data in a contrastive manner, Zhang *et al.* [ZJG19] propose a contrast module, Zhang *et al.* [ZZW19] formulate it in a student-teacher setting, Wang *et al.* [WJL20] build a multiplex graph network, Hu *et al.* [HML20] aggregate features from a hierarchical decomposition, and Wu *et al.* [WDG20] apply a scattering transformation to learn objects, attributes, and relations. Recently, Zhang *et al.* [ZJZ21] employ a neuro-symbolic design but requires full knowledge over the hidden relations to perform *abduction*. While our work adopts the visual perception module and employs a similar training strategy from Zhang *et al.* [ZJZ21], the ALANS learner manages to *induce* the hidden relations, enabling on-the-fly relation induction and systematic generalization on relational learning. The recent work of Neural Interpreter (NI) [RGJ21] is a complementary neural approach to our method: Although both NI and ALANS decompose the reasoning process into sub-components and aggregate them, NI focuses more on compositionality, routing new input via different paths of learned modules to generalize, whereas ALANS more on induction, enabling a learned module to adapt on the fly.

5.3 The ALANS Learner

In this section, we introduce the ALANS learner for the RPM problem. In each RPM instance, an agent is given an incomplete 3×3 panel matrix with the last entry missing and asked to induce the operator hidden in the matrix and choose from eight choice panels one that follows it. Formally, let the answer variable be denoted as y, the context panels as $\{I_{o,i}\}_{i=1}^{8}$, and choice panels as $\{I_{c,i}\}_{i=1}^{8}$. Then the problem can be formulated as estimating $P(y \mid \{I_{o,i}\}_{i=1}^{8}, \{I_{c,i}\}_{i=1}^{8})$. According to the common design [CJS90, SHB18, ZGJ19], there is one operator that governs each panel attribute. Hence, by assuming independence among attributes, we propose to factorize the probability of $P(y = n \mid \{I_{o,i}\}_{i=1}^{8}, \{I_{c,i}\}_{i=1}^{8})$ as

$$\prod_{a} \sum_{\mathcal{T}^a} P(y^a = n \mid \mathcal{T}^a, \{I_{o,i}\}_{i=1}^{8}, \{I_{c,i}\}_{i=1}^{8}) \times P(\mathcal{T}^a \mid \{I_{o,i}\}_{i=1}^{8}), \tag{5.1}$$

where y^a denotes the answer selection based only on attribute a and $\mathcal{T}^a$ the operator on a.

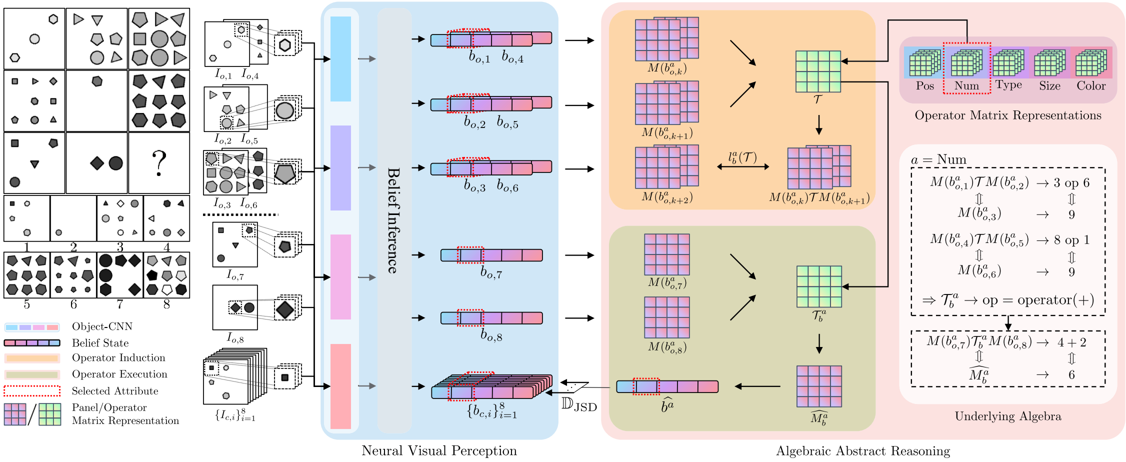


Figure 5.1: **An overview of the ALANS learner.** For an RPM instance, the neural visual perception module produces the belief states for all panels: an object CNN extracts object attribute distributions for each image region, and a belief inference engine marginalizes them out to obtain panel attribute distributions. For each panel attribute, the algebraic abstract reasoning module transforms the belief states into matrix-based algebraic representation and induces hidden operators by solving inner optimizations. The answer representation is obtained by executing the induced operators, and the choice most similar to the prediction is selected as the solution. An example of the underlying discrete algebra and its correspondence is also shown on the right.

Overview As shown in Fig. 5.1, the ALANS learner decomposes the process into perception and reasoning: the neural visual perception frontend is adopted from Zhang *et al*. [ZJZ21] and extracts the *belief states* from each of the sixteen panels, whereas the algebraic abstract reasoning backend views an instance as an example in an abstract algebra structure, transforms belief states into *algebraic representation* by the representation theory, *induces* the hidden operators, and *executes* the operators to predict the representation of the answer. Therefore, in Eq. (5.1), the operator distribution is modeled by the fitness of an operator and the answer distribution by the distance between the predicted representation and that of a candidate.

5.3.1 Neural Visual Perception

We follow the design in Zhang *et al.* [ZJZ21] and decompose visual perception into an object CNN and a belief state inference engine. Specifically, for each panel, we use a sliding window to traverse the spatial domain of the image and feed each image region into an object CNN. The CNN has four branches, producing for each region its object attribute distributions, including objectiveness (if the region contains an object), type, size, and color. The belief inference engine summarizes the panel attribute distributions (over position, number, type, size, and color) by marginalizing out all object attribute distributions (over objectiveness, type, size, and color). As an example, the distribution of the panel attribute of Number can be computed as such: for N image regions and their predicted objectiveness

$$P(\text{Number} = k) = \sum_{\substack{R^o \in \{0,1\}^N \\ \sum_j R^o_j = k}} \prod_{j=1}^{N} P(r^o_j = R^o_j), \tag{5.2}$$

where $P(r^o_j)$ denotes the jth region's estimated objectiveness distribution, and R^o is a binary sequence of length N that sums to k. All panel attribute distributions compose the *belief state* of a panel. In the following, we denote the belief state as b and the distribution of an attribute a as $P(b^a)$. For more details, please refer to Zhang *et al.* [ZJZ21].

5.3.2 Algebraic Abstract Reasoning

Given the belief states of both context and choice panels, the algebraic abstract reasoning backend concerns the induction of hidden operators and the prediction of answer representation for each attribute. The fitness of induced operators is used for estimating the operator distribution and the difference between the prediction and the choice panel for estimating the answer distribution.

Algebraic Underpinning Without loss of generality, here we assume row-wise operators. For each attribute, under perfect perception, the first two rows in an RPM instance provide

$$\begin{tikzcd}
M(b^a_{o,i}) \arrow[rr, bend left] \arrow[d, shift left, "M^{-1}(\cdot)"] & M(b^a_{o,i+1}) \arrow[r, bend left, "\mathcal{T}"] \arrow[d, shift left, "M^{-1}(\cdot)"] & M(b^a_{o,i+2}) \arrow[d, shift left, "M^{-1}(\cdot)"] \\
b^a_{o,i} \arrow[u, shift left, "M(\cdot)"] \arrow[rr, bend right] & b^a_{o,i+1} \arrow[u, shift left, "M(\cdot)"] \arrow[r, bend right, "?"'] & b^a_{o,i+2} \arrow[u, shift left, "M(\cdot)"]
\end{tikzcd}$$

Figure 5.2: **Isomorphism between the abstract algebra and the matrix-based representation.** In this view, operator induction is now reduced to solving for a matrix.

snapshots into an example of *group* [HO37] constrained to an integer-indexed set, a simple algebra structure that is closed under a binary operator. To see this, note that an accurate perception module would see each panel attribute as a deterministic set element. Therefore, RPM instances with unary operators, such as progression, are group examples with special binary operators where one operand is constant. Instances with binary operators, such as arithmetics, directly follow the group properties. Those with ternary operators are ones defined on a three-tuple set from rows.

Algebraic Representation A systematic algebraic view allows us to felicitously recruit ideas in the representation theory [Hum12] to glean the hidden properties in the abstract structures: it makes abstract algebra amenable by reducing it onto linear algebra. Following the same spirit, we propose to lift both the set elements and the hidden operators to a learnable matrix space. To encode the set element, we employ the Peano Axiom [Pea89]. According to the Peano Axiom, an integer-indexed set can be constructed by (1) a zero element ($\mathbf{0}$), (2) a successor function ($S(\cdot)$), and (3) the principle of mathematical induction, such that the kth element is encoded as $S^k(\mathbf{0})$. Specifically, we instantiate the zero element as a learnable matrix M_0 and the successor function as the matrix-matrix product parameterized by M. In an attribute-specific manner, the representation of an attribute taking the kth value is $(M^a)^k M_0^a$. For operators, we consider them to live in a learnable matrix group of a corresponding dimension, such that the action of an operator on a set can be represented as matrix multiplication. Such algebraic representation establishes an isomorphism between the matrix space and the abstract algebraic structure: abstract elements on the algebraic structure have a bijective mapping to/from the matrix space, and inducing the abstract relation can be reduced to solving for a matrix operator. See Fig. 5.2 for a graphical illustration of the isomorphism.

Operator Induction Operator induction concerns about finding a concrete operator in the abstract algebraic structure. By the property of closure, we formulate it as an inner-level regularized linear regression problem: a binary operator $\mathcal{T}_b^a$ for attribute a in a group minimizes $\ell_b^a(\mathcal{T})$ defined as

$$\ell_b^a(\mathcal{T}) = \sum_i \mathbb{E}\left[\|M(b_{o,i}^a)\mathcal{T}M(b_{o,i+1}^a) - M(b_{o,i+2}^a)\|_F^2\right] + \lambda_b^a \|\mathcal{T}\|_F^2, \tag{5.3}$$

where under visual uncertainty, we take the expectation w.r.t. the distributions in the belief states of context panels $P(b_{o,i}^a)$ in the first two rows, and denote its algebraic representation as $M(b_{o,i}^a)$. For unary operators, one operand can be treated as constant and absorbed into $\mathcal{T}$. Note that Eq. (5.3) admits a closed-form solution. Therefore, the operator can be learned and adapted for different instances of binary relations and concluded on the fly. Such a design also simplifies the recent neuro-symbolic approaches, where every single symbol operator needs to be hand-defined [HMG19, MGK19, YGL20, YWG18]. Instead, we only specify an inner-level optimization framework and allow symbolic operators to be quickly induced based on the neural observations, while keeping the semantic interpretability in the neuro-symbolic methods. Therefore, we term such a design semi-symbolic.

The operator probability in Eq. (5.1) is then modeled by each operator type's fitness, *e.g.*, for binary,

$$P(\mathcal{T}^a = \mathcal{T}_b^a \mid \{I_{o,i}\}_{i=1}^8) \propto \exp(-\ell_b^a(\mathcal{T}_b^a)). \tag{5.4}$$

Operator Execution To predict the algebraic representation of the answer, we solve another inner-level optimization similar to Eq. (5.3), but now treating the representation of the answer as a variable:

$$\widehat{M_b^a} = \arg\min_M \ell_b^a(M) = \mathbb{E}[\|M(b_{o,7}^a)\mathcal{T}_b^a M(b_{o,8}^a) - M\|_F^2], \tag{5.5}$$

where the expectation is taken w.r.t. context panels in the last row. The optimization also admits a closed-form solution, which corresponds to the execution of the induced operator in Eq. (5.3).

The predicted representation is decoded probabilistically as the predicted belief state of the solution,

$$P(\widehat{b^a} = k \mid \mathcal{T}^a) \propto \exp(-\|\widehat{M^a} - (M^a)^k M_0^a\|_F^2). \tag{5.6}$$

Answer Selection Based on Eqs. (5.1) and (5.4), estimating the answer distribution is now boiled down to estimating the conditional answer distributions for each attribute. Here, we propose to model it based on the JSD of the predicted belief state and that of a choice,

$$P(y^a = n \mid \mathcal{T}^a, \{I_{o,i}\}_{i=1}^8, \{I_{c,i}\}_{i=1}^8) \propto \exp(-d_n^a), \tag{5.7}$$

where we define d_n^a as

$$d_n^a = \mathbb{D}_{\text{JSD}}(P(\widehat{b^a} \mid \mathcal{T}^a) \| P(b_{c,n}^a))). \tag{5.8}$$

Discussion Comparing with the possible problem-solving process by humans [CJS90], we argue that the proposed algebraic abstract reasoning module offers a computational and interpretable counterpart to human-like reasoning in RPM. Specifically, the induction component resembles fluid intelligence, where one quickly induces the hidden operator by observing the context panels. The execution component synthesizes an image by executing the induced operator, and the choice most similar to the image is selected as the answer.

We also note that by decoding the predicted representation in Eq. (5.6), a solution can be *generated*: by sequentially selecting the most probable operator and the most probable attribute value, a rendering engine can directly render the solution. The reasoning backend also enables end-to-end training: by integrating the belief states from neural perception, the module conducts both induction and execution in a soft manner, such that the gradients can be back-propagated and both the visual frontend and the reasoning backend jointly trained.

5.3.3 Training Strategy

We train the entire ALANS learner by minimizing the cross-entropy loss between the estimated answer distribution and the ground-truth selection and an auxiliary loss [SHB18, WJL20, ZGJ19, ZJZ21] that shapes the operator distribution from the reasoning engine, *i.e.*,

$$\min_{\theta,\{M_0^a\},\{M^a\}} \ell(P(y \mid \{I_{o,i}\}_{i=1}^8, \{I_{c,i}\}_{i=1}^8), y_\star) + \sum_a \lambda^a \ell(P(\mathcal{T}^a \mid \{I_{o,i}\}_{i=1}^8), y_\star^a), \tag{5.9}$$

where $\ell(\cdot)$ denotes the cross-entropy loss, $y_\star$ the correct choice in candidates, and $y_\star^a$ the ground-truth operator selection for attribute a. The first part of the loss encourages the model to select the right choice for evaluation, while the second part motivates meaningful internal representation to emerge. Compared to Zhang *et al.* [ZJZ21], the system requires joint operation from not only a trained perception module θ, but also the algebraic encodings from the zero elements $\{M_0^a\}$ and the successor functions $\{M^a\}$, and correspondingly, induced operators $\mathcal{T}$. We notice the three-stage curriculum in Zhang *et al.* [ZJZ21] is crucial for such a neuro-semi-symbolic system. In particular, we use λ^a to balance the trade-off in the curriculum: in the first stage, we only train parameters regarding objectiveness; in the second stage, we freeze objectiveness parameters and cyclically train parameters involving type, size, and color; in the last stage, we fine-tune all parameters.

5.4 Experiments

A cognitive architecture with systematic generalization is believed to demonstrate the following three principles [FP88, Mar01, Mar20]: (1) systematicity, (2) productivity, and (3) localism. Systematicity requires an architecture to be able to entertain “semantically related” contents after understanding a given thought. Productivity states the awareness of a constituent implies that of a recursive application of the constituent; vice versa for localism.

To verify the effectiveness of an algebraic treatment in systematic generalization, we showcase the superiority of the proposed ALANS learner on the three principles in the abstract spatial-temporal reasoning task of RPM. Specifically, we use the generation methods proposed in Zhang *et al*. [ZGJ19] and Hu *et al*. [HML20] to generate RPM problems and carefully split training and testing to construct the three regimes. The former generates candidates by perturbing only one attribute of the correct answer while the later modifies attribute values in a hierarchical manner to avoid shortcut solutions by pure statistics. Both methods categorize relations in RPM into three types, according to Carpenter *et al*. [CJS90]: unary (Constant and Progression), binary (Arithmetic), and ternary (Distribution of Three), each of which comes with several instances. Grounding the principles into learning abstract relations in RPM, we fix the configuration to be 3×3Grid and generate the following data splits for evaluation:

- Systematicity: the training set contains only a subset of instances for each type of relation, while the test set all other relation instances.
- Productivity: as the binary relation results from a recursive application of the unary relation, the training set contains only unary relations, whereas the test set only binary relations.
- Localism: the training and testing sets in the productivity split are swapped to study localism.

We follow Zhang *et al*. [ZGJ19] to generate $10,000$ instances for each split and assign 6 folds for training, 2 folds for validation, and 2 folds for testing.

5.4.1 Experimental Setup

We evaluate the systematic generalizability of the proposed ALANS learner on the above three splits, and compare the ALANS learner with other baselines, including ResNet [HZR16], ResNet+DRT [ZGJ19], WReN [SHB18], CoPINet [ZJG19], MXGNet [WJL20], LEN [ZZW19], HriNet [HML20], and SCL [WDG20]. We use either official or public implementations that reproduce the original results. All models are implemented in PyTorch [PGC17] and optimized using ADAM [KB14] on an Nvidia Titan Xp GPU. We validate trained models on validation sets and report performance on test sets.

5.4.2 Systematic Generalization

Tab. 5.1 shows the performance of various models on systematic generalization, *i.e.*, systematicity, productivity, and localism. Compared to results reported in existing works mentioned above, all pure connectionist models experience a devastating performance drop when it comes to the critical cognitive requirements on systematic generalization, indicating that pure connectionist models fail to perform abstraction, algebraization, induction, or generalization needed in solving the abstract reasoning task; instead, they seem to only take a shortcut to bypass them. In particular, MXGNet's [WJL20] superiority is diminishing in systematic generalization. In spite of learning with structural annotations, ResNet+DRT [ZGJ19] does not fare better than its base model. The recently proposed HriNet [HML20] slightly improves on ResNet [HZR16] in this aspect, with LEN [ZZW19] being only marginally better. WReN [SHB18], on the other hand, shows oscillating performance across three regimes. Evaluated under systematic generation, SCL [WDG20] and CoPINet [ZJG19] also far deviate from "superior performance." These observations suggest that pure connectionist models highly likely learn from variation in visual appearance rather than the algebra underlying the problem.

Embedded in a neural-semi-symbolic framework, the proposed ALANS learner improves

Method	MXGNet	ResNet+DRT	ResNet	HriNet	LEN	WReN	SCL	CoPINet	ALANS	ALANS-Ind	ALANS-V
Systematicity	20.95%	33.00%	27.35%	28.05%	40.15%	35.20%	37.35%	59.30%	**78.45**%	52.70%	93.85%
Productivity	30.40%	27.95%	27.05%	31.45%	42.30%	56.95%	51.10%	60.00%	**79.95**%	36.45%	90.20%
Localism	28.80%	24.90%	23.05%	29.70%	39.65%	38.70%	47.75%	60.10%	**80.50**%	59.80%	95.30%
Average	26.72%	28.62%	25.82%	29.73%	40.70%	43.62%	45.40%	59.80%	**79.63**%	48.65%	93.12%
Systematicity	13.35%	13.50%	14.20%	21.00%	17.40%	15.00%	24.90%	18.35%	**64.80**%	52.80%	84.85%
Productivity	14.10%	16.10%	20.70%	20.35%	19.70%	17.95%	22.20%	29.10%	**65.55**%	32.10%	86.55%
Localism	15.80%	13.85%	17.45%	24.60%	20.15%	19.70%	29.95%	31.85%	**65.90**%	50.70%	90.95%
Average	14.42%	14.48%	17.45%	21.98%	19.08%	17.55%	25.68%	26.43%	**65.42**%	45.20%	87.45%

Table 5.1: **Model performance on different aspects of systematic generalization.** The performance is measured by accuracy on the test sets. Results on datasets generated by Zhang *et al.* [ZGJ19] (upper) and by Hu *et al.* [HML20] (lower).

on systematic generalization by a large margin. With an algebra-aware design, the model is considerably stable across different principles of systematic generalization. The algebraic representation learned in relations of either a constituent or a recursive composition naturally supports productivity and localism, while semi-symbolic inner optimization further allows various instances of an operator type to be induced from the algebraic representation and boosts systematicity. The importance of the algebraic representation is made more significant in the ablation study: ALANS-Ind, with algebraic representation replaced by independent encodings and the algebraic isomorphism broken, shows inferior performance. We also examine the performance of the learner with perfect visual annotations (denoted as ALANS-V) to see how the proposed algebraic reasoning module works: the gap despite of accurate perception indicates space for improvement for the inductive reasoning part of the model. In the next section, we further show that the neuro-semi-symbolic decomposition in ALANS's design enables diagnostic tests into its jointly learned perception module and reasoning module. This design is in stark contrast to black-box models.

5.4.3 Analysis into Perception and Reasoning

The neural-semi-symbolic design affords analyses into both perception and reasoning. To evaluate the neural perception and the algebraic reasoning modules, we extract region-based object attribute annotations from the datasets [HML20, ZGJ19] and categorize all relations into three types, *i.e.*, unary, binary, and ternary.

Tab. 5.2 shows the perception module's performance on the test sets in the three regimes of systematic generalization. We note that in order for the ALANS learner to achieve the desired results shown in Tab. 5.1, ALANS learns to construct the concept of objectiveness perfectly. The model also shows fairly accurate prediction on the attributes of type and size. However, on the texture-related concept of color, ALANS fails to develop a reliable notion on it. Despite that, the general prediction accuracy of the perception module is still surprising, considering that the perception module is jointly learned with ground-truth annotations on answer selections. The relatively lower accuracy on color could be attributed to its larger space compared to other attributes.

Object Attribute	Objectiveness	Type	Size	Color
Systematicity	100.00%	99.95%	94.65%	71.35%
Productivity	100.00%	99.97%	98.04%	77.61%
Localism	100.00%	95.65%	98.56%	80.05%
Average	100.00%	98.52%	97.08%	76.34%

Object Attribute	Objectiveness	Type	Size	Color
Systematicity	100.00%	96.34%	92.36%	63.98%
Productivity	100.00%	94.28%	97.00%	69.89%
Localism	100.00%	95.80%	98.36%	60.35%
Average	100.00%	95.47%	95.91%	64.74%

Table 5.2: **Perception accuracy of the proposed ALANS learner, measured by whether the module can correctly predict an attribute's value.** Results on datasets generated by Zhang *et al.* [ZGJ19] (left) and by Hu *et al.* [HML20] (right).

Tab. 5.3 lists the reasoning module's performance during testing for the three aspects. Note that on position, the unary operator (shifting) and binary operator (set arithmetics) do not systematically imply each other. Hence, we do not count them as probes into productivity and localism. In general, we notice that the better the perception accuracy on one attribute, the better the performance on reasoning. However, we also note that despite the relatively accurate perception of objectiveness, type, and size, near perfect reasoning is never guaranteed. This deficiency is due to the perception uncertainty handled by expectation in Eq. (5.3): in spite of correctness when we take arg max, marginalizing by expectation will unavoidably introduce noise into the reasoning process. Therefore, an ideal reasoning module requires the perception frontend to be not only correct but also certain. Computationally, one can sample from the perception module and optimize Eq. (5.9) using REINFORCE [Wil92]. However, the credit assignment problem and variance in gradient estimation will further complicate training.

Relation on	Position	Number	Type	Size	Color
Systematicity	72.04%	82.14%	81.50%	80.80%	40.40%
Productivity	-	98.75%	89.50%	72.10%	33.95%
Localism	-	74.70%	44.25%	56.40%	54.20%
Average	72.04%	85.20%	71.75%	69.77%	42.85%

Relation on	Position	Number	Type	Size	Color
Systematicity	69.96%	80.34%	83.50%	80.85%	28.85%
Productivity	-	99.10%	87.95%	68.50%	23.10%
Localism	-	70.55%	36.65%	42.30%	33.20%
Average	69.96%	83.33%	69.37%	63.88%	28.38%

Table 5.3: **Reasoning accuracy of the proposed ALANS learner, measured by whether the module can correctly predict the type of a relation on an attribute.** Results on datasets generated by Zhang *et al.* [ZGJ19] (left) and by Hu *et al.* [HML20] (right).

5.4.4 In-Distribution Performance

To further evaluate how models perform under the regular I.I.D. setup, we train the models on the original datasets generated by Zhang *et al.* [ZGJ19] and Hu *et al.* [HML20] and measure the model accuracy in the test splits. We compare ALANS with published baselines in Tab. 5.1.

Tab. 5.4 (left) shows the results on the RAVEN dataset [ZGJ19]. With the jointly trained vision component, the ALANS learner does not fare better than the best connectionist approaches, making it on par with SCL only. As the dataset is known to have shortcut solutions, neural approaches like MXGNet and CoPINet could potentially find it easier to solve and

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
WReN	34.0%/21.5%	58.4%/24.0%	38.9%/25.0%	37.7%/20.1%	21.6%/19.7%	19.8%/19.9%	38.9%/21.3%	22.6%/20.6%
ResNet	53.4%/18.4%	52.8%/22.6%	41.9%/15.5%	44.3%/18.1%	58.8%/19.0%	60.2%/19.6%	63.2%/17.5%	53.1%/16.6%
ResNet+DRT	59.6%/20.7%	58.1%/24.2%	46.5%/18.2%	50.4%/19.8%	65.8%/22.0%	67.1%/22.1%	69.1%/21.0%	60.1%/18.1%
LEN	71.6%/32.8%	79.1%/44.8%	56.1%/27.9%	60.3%/23.9%	80.5%/34.1%	76.4%/34.4%	79.3%/35.8%	69.9%/28.5%
HriNet	45.1%/60.8%	66.1%/78.2%	40.7%/50.1%	38.0%/42.4%	44.9%/70.1%	43.2%/70.3%	47.2%/68.2%	35.8%/46.3%
MXGNet	84.0%/33.1%	94.3%/40.7%	60.5%/27.9%	64.9%/24.7%	96.6%/35.8%	96.4%/34.5%	94.1%/36.4%	81.3%/31.6%
CoPINet	91.4%/46.1%	95.1%/54.4%	77.5%/36.8%	78.9%/31.9%	**99.1**%/51.9%	**99.7**%/52.5%	**98.5**%/52.2%	91.4%/42.8%
ALANS	74.4%/78.5%	69.1%/72.3%	80.2%/79.5%	75.0%/72.9%	72.2%/79.2%	73.3%/79.6%	76.3%/85.9%	74.9%/79.9%
SCL	74.2%/80.5%	82.8%/84.6%	70.4%/79.4%	64.1%/69.9%	77.6%/82.7%	78.4%/82.6%	84.2%/87.3%	62.2%/77.2%
ALANS-V	**94.4**%/**93.5**%	**98.4**%/**98.9**%	**91.5**%/**85.0**%	**87.0**%/**83.2**%	97.3%/**90.9**%	96.4%/**98.1**%	97.3%/**99.1**%	**93.2**%/**89.5**%

Table 5.4: **Model performance on RAVEN [ZGJ19] (left) and I-RAVEN [HML20] (right) under the regular I.I.D. evaluation, measured by accuracy on the test sets.**

hence achieve much superior results in this setup. However, ALANS-V, the variant with a perfect perception component reaches a level of much robustness and accuracy, attaining the best results in grid-like layouts, empirically believed to be the hardest in human evaluation [ZGJ19].

Tab. 5.4 (right) shows the results on the I-RAVEN dataset [HML20]. Apart from ALANS-V's realizing the best performance across all models, we also notice the consistency of performance of the proposed method across datasets, with or without the shortcut issues. All other methods show drastically varying performance, particularly for CoPINet and MXGNet, arguably because of the choice generation strategy that effectively prunes easy solution paths via statistics.

In summary, by analyzing the results from Tabs. 5.1 and 5.4 together, we notice that ALANS not only attains reasonable performance on the I.I.D. setup but also generalizes systematically.

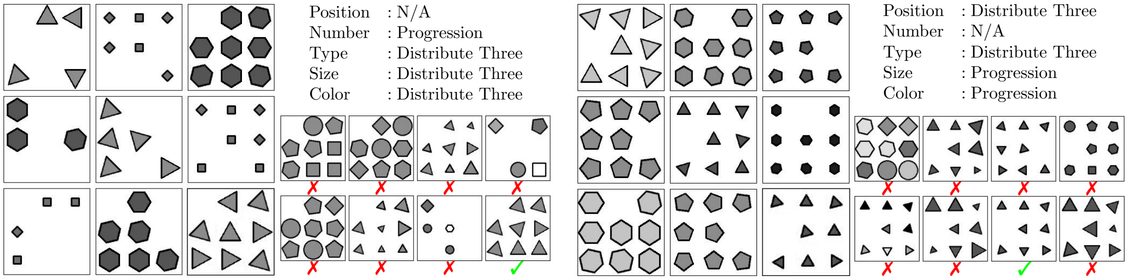


Figure 5.3: **Examples of RPM instances with the missing entries filled by solutions directly generated by the ALANS learner.** Ground-truth relations are also listed. Note the generated results do not look exactly the same as the correct candidate choices due to random rotations during rendering, but they are semantically correct.

5.4.5 Generative Potential

Compared to existing discriminative-only RPM-solving methods, the proposed ALANS learner is unique in its generative potential. As mentioned above, the final panel attribute can be decoded by sequentially selecting the most probable hidden operator and the attribute value. A solution can be generated when equipped with a rendering engine. In Fig. 5.3, we use the rendering program from Zhang *et al*. [ZGJ19] to showcase the generative potential in the ALANS learner.

5.5 Conclusion and Limitation

In this work, we propose the ALgebra-Aware Neuro-Semi-Symbolic (ALANS) learner, echoing a normative theory in the connectionist-classicist debate that an algebraic treatment in a cognitive architecture should improve a model's systematic generalization ability. In particular, the ALANS learner employs a neural-semi-symbolic architecture, where the neural visual perception module is responsible for summarizing visual information and the algebraic abstract reasoning module transforms it into algebraic representation with isomorphism established by the Peano Axiom and the representation theory, conducts operator induction, and executes it to arrive at an answer. In three RPM domains reflective of systematic generalization, the proposed ALANS learner shows superior performance compared to other pure connectionist baselines.

The proposed ALANS learner also bears some limitations. For one thing, we make the assumption in our formulation that relations on different attributes are independent and can be factorized. This assumption is not universally correct and could potentially lead to failure in more complex reasoning scenarios when attributes are correlated. For another, we assume a fixed and known space for each attribute in the perception module, while in the real world the space for one attribute could be dynamically changing. In addition, the reasoning module is sensitive to perception uncertainty as has already been discussed in the experimental results. Besides, the gap between perfection and the status quo in reasoning remains to be filled. In this work, we only show how the hidden operator can be induced with regularized linear regression via the representation theory. However, more elaborate differentiable optimization problems can certainly be incorporated for other problems.

With the limitation in mind, we hope that this preliminary study could inspire more research on incorporating algebraic structures into current connectionist models and help address challenging modeling problems [XJZ22, ZJE21, ZZZ20, ZGF20].

CHAPTER 6

Conclusion

In this dissertation, we detail the series of exploration we made to equip machines with the ability of few-shot concept induction. In particular, we start from the introduction of a comprehensive evaluation benchmark, to data-driven contrastive learning, and finally landing in neuro-symbolic and neuro-semi-symbolic methods.

In the exploration, we note that despite of learning contrastively, data-driven methods still tend to capture the statistical correlation in the dataset rather than mastering how to perform efficient few-shot concept induction. While providing data that covers the entire space could be a straightforward solution, its sheer volume and labeling cost are nothing to be neglected. Besides, we can't expect a model learned in this way to generalize as humans.

Neuro-symbolic and neuro-semi-symbolic methods point out potential benefits of such learning frameworks for few-shot concept induction. By disentangling perception and reasoning, these frameworks can seamlessly integrate classic reasoning methods like planning and optimization in the process of few-shot concept induction. However, we also note that these systems require more comprehensive scene understanding and structured modeling of the problem. Fortunately, the former can be partially achieved by recent parsing methods. Yet, finding a uniform space for relational modeling is non-trivial.

Despite of the challenges, we hope that this preliminary study shows the superiority of coupling structured methods with recent data-driven learning and could inspire more research in leveraging classic reasoning methods, like planning, inverse planning, optimization, recursive representation, to address challenging problems facing the entire community.

Bibliography

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering." In *International Conference on Computer Vision (ICCV)*, 2015.

[Ams62] Abram Amsel. "Frustrative nonreward in partial reinforcement and discrimination learning: Some recent history and a theoretical extension." *Psychological review*, **69**(4):306, 1962.

[Arn69] Rudolf Arnheim. *Visual thinking*. Univ of California Press, 1969.

[Aug76] Saint Augustine. *The confessions*. Clark, 1876.

[AYB18] Somak Aditya, Yezhou Yang, and Chitta Baral. "Explicit Reasoning over End-to-End Neural Architectures for Visual Question Answering." *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[Bar13] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[BHS18] David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. "Measuring abstract reasoning in neural networks." In *International Conference on Machine Learning (ICML)*, pp. 511–520, 2018.

[BLC09] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. "Curriculum learning." In *International Conference on Machine Learning (ICML)*, 2009.

[BMN19] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. "Systematic Generalization: What Is Required and Can It Be Learned?" In *International Conference on Learning Representations (ICLR)*, 2019.

[Bow61] Gordon H Bower. “A contrast effect in differential conditioning.” *Journal of Experimental Psychology*, **62**(2):196, 1961.

[BSC18] Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. “Learning Interpretable Spatial Operations in a Rich 3D Blocks World.” *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[CFH92] David J Chalmers, Robert M French, and Douglas R Hofstadter. “High-level perception, representation, and analogy: A critique of artificial intelligence methodology.” *Journal of Experimental & Theoretical Artificial Intelligence*, **4**(3):185–211, 1992.

[CH89] Richard Catrambone and Keith J Holyoak. “Overcoming contextual limitations on problem-solving transfer.” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **15**(6):1147, 1989.

[CHL05] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. “Learning a similarity metric discriminatively, with application to face verification.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[Cho19] François Chollet. “The Measure of Intelligence.” *arXiv preprint arXiv:1911.01547*, 2019.

[CHY19] Yixin Chen, Siyuan Huang, Tao Yuan, Yixin Zhu, Siyuan Qi, and Song-Chun Zhu. “Holistic++ Scene Understanding: Single-view 3D Holistic Scene Parsing and Human Pose Estimation with Human-Object Interaction and Physical Commonsense.” In *International Conference on Computer Vision (ICCV)*, 2019.

[CJS90] Patricia A Carpenter, Marcel A Just, and Peter Shell. “What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test.” *Psychological review*, **97**(3):404, 1990.

[CLL18] Qingxing Cao, Xiaodan Liang, Bailing Li, Guanbin Li, and Liang Lin. "Visual Question Reasoning on General Dependency Tree." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[CLY20] Xinyun Chen, Chen Liang, Adams Wei Yu, Dawn Song, and Denny Zhou. "Compositional generalization via neural-symbolic stack machines." In *Advances in Neural Information Processing Systems*, 2020.

[CM09] Kate Crookes and Elinor McKone. "Early maturity of face recognition: No childhood development of holistic processing, novel face encoding, or face-space." *Cognition*, **111**(2):219–247, 2009.

[CMS07] Benoît Colson, Patrice Marcotte, and Gilles Savard. "An overview of bilevel optimization." *Annals of Operations Research*, **153**(1):235–256, 2007.

[CMW20] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. "Grounding Physical Concepts of Objects and Events Through Dynamic Visual Reasoning." In *International Conference on Learning Representations (ICLR)*, 2020.

[CR68] Fergus W Campbell and JG Robson. "Application of Fourier analysis to the visibility of gratings." *The Journal of physiology*, **197**(3):551–566, 1968.

[CSS10] Catherine C Chase, Jonathan T Shemwell, and Daniel L Schwartz. "Explaining across contrasting cases for deep understanding in science: An example using interactive simulations." In *Proceedings of the 9th International Conference of the Learning Sciences*, 2010.

[DIP06] Stanislas Dehaene, Véronique Izard, Pierre Pica, and Elizabeth Spelke. "Core knowledge of geometry in an Amazonian indigene group." *Science*, **311**(5759):381–384, 2006.

[DL17] Bo Dai and Dahua Lin. "Contrastive learning for image captioning." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[DML18] Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. "Neural Logic Machines." In *International Conference on Learning Representations (ICLR)*, 2018.

[EG18] Richard Evans and Edward Grefenstette. "Learning explanatory rules from noisy data." *Journal of Artificial Intelligence Research (JAIR)*, **61**:1–64, 2018.

[EKM84] R E Snow, Patrick Kyllonen, and B Marshalek. "The topography of ability and learning correlations." *Advances in the psychology of human intelligence*, pp. 47–103, 1984.

[EKS18] Mark Edmonds, Feng Kubricht, James, Colin Summers, Yixin Zhu, Brandon Rothrock, Song-Chun Zhu, and Hongjing Lu. "Human Causal Transfer: Challenges for Deep Reinforcement Learning." In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2018.

[EMQ20] Mark Edmonds, Xiaojian Ma, Siyuan Qi, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. "Theory-based Causal Transfer: Integrating Instance-level Induction and Abstract-level Structure Learning." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[EQZ19] Mark Edmonds, Siyuan Qi, Yixin Zhu, James Kubricht, Song-Chun Zhu, and Hongjing Lu. "Decomposing Human Causal Learning: Bottom-up Associative Learning and Top-down Schema Reasoning." In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2019.

[Eva62] TG Evans. *A Heuristic Program to Solve Geometric Analogy Problems.* PhD thesis, MIT, 1962.

[Eva64] Thomas G Evans. "A heuristic program to solve geometric-analogy problems." In *Proceedings of the April 21-23, 1964, spring joint computer conference*, 1964.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In *International Conference on Machine Learning (ICML)*, 2017.

[Fod75] Jerry A Fodor. *The language of thought*, volume 5. Harvard university press, 1975.

[FP88] Jerry A Fodor, Zenon W Pylyshyn, et al. "Connectionism and cognitive architecture: A critical analysis." *Cognition*, **28**(1-2):3–71, 1988.

[Fu74] King Sun Fu. *Syntactic methods in pattern recognition*, volume 112. Elsevier, 1974.

[Gal83] Francis Galton. *Inquiries into human faculty and its development.* Macmillan, 1883.

[GBG12] Artur S d'Avila Garcez, Krysia B Broda, and Dov M Gabbay. *Neural-symbolic learning systems: foundations and applications.* Springer Science & Business Media, 2012.

[Gen83] Dedre Gentner. "Structure-mapping: A theoretical framework for analogy." *Cognitive science*, **7**(2):155–170, 1983.

[GG55] James J Gibson and Eleanor J Gibson. "Perceptual learning: Differentiation or enrichment?" *Psychological review*, **62**(1):32, 1955.

[GG01] Dedre Gentner and Virginia Gunn. "Structural alignment facilitates the noticing of differences." *Memory & Cognition*, **29**(4):565–577, 2001.

[GH10] Michael Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[Gib14] James J Gibson. *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

[GM94] Dedre Gentner and Arthur B Markman. "Structural alignment in comparison: No difference without similarity." *Psychological science*, **5**(3):152–158, 1994.

[GNT04] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice.* Elsevier, 2004.

[GP92] Mary L Gick and Katherine Paterson. "Do contrasting examples facilitate schema acquisition and analogical transfer?" *Canadian Journal of Psychology/Revue canadienne de psychologie*, **46**(4):539, 1992.

[Gra06] Temple Grandin. *Thinking in pictures: And other reports from my life with autism.* Vintage, 2006.

[Gre76] Ulf Grenander. "Lectures in pattern theory I, II and III: Pattern analysis, pattern synthesis and regular structures.", 1976.

[GZW07] Cheng-en Guo, Song-Chun Zhu, and Ying Nian Wu. "Primal sketch: Integrating structure and texture." *Computer Vision and Image Understanding (CVIU)*, **106**(1):5–19, 2007.

[HAR17] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. "Learning to Reason: End-to-End Module Networks for Visual Question Answering." In *International Conference on Computer Vision (ICCV)*, 2017.

[HDW09] Rubi Hammer, Gil Diesendruck, Daphna Weinshall, and Shaul Hochstein. "The development of category learning strategies: What makes the difference?" *Cognition*, **112**(1):105–119, 2009.

[Hea56] Thomas Little Heath et al. *The thirteen books of Euclid's Elements*. Courier Corporation, 1956.

[HHT96] Keith J Holyoak, Keith James Holyoak, and Paul Thagard. *Mental leaps: Analogy in creative thought*. MIT press, 1996.

[HIL22] Keith J Holyoak, Nicholas Ichien, and Hongjing Lu. "From Semantic Vectors to Analogical Mapping." *Current Directions in Psychological Science*, p. 09637214221098054, 2022.

[HIO11] Etsuko Haryu, Mutsumi Imai, and Hiroyuki Okada. "Object similarity bootstraps young children to action-based verb extension." *Child Development*, **82**(2):674–686, 2011.

[HM12] Keith James Holyoak and Robert G Morrison. *The Oxford handbook of thinking and reasoning*. Oxford University Press, 2012.

[HM18] Drew A Hudson and Christopher D Manning. "Compositional attention networks for machine reasoning." *arXiv preprint arXiv:1803.03067*, 2018.

[HM19] Drew Hudson and Christopher D Manning. "Learning by abstraction: The neural state machine." In *Advances in Neural Information Processing Systems*, 2019.

[HMG19] Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. "Visual Concept-Metaconcept Learning." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[HML20] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. "Hierarchical Rule Induction Network for Abstract Visual Reasoning." *arXiv preprint arXiv:2002.06838*, 2020.

[HML21] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. "Stratified Rule-Aware Network for Abstract Visual Reasoning." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[HNF19] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. "Divergence Triangle for Joint Training of Generator Model, Energy-based Model, and Inferential Model." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[HO37] Bernard A Hausmann and Oystein Ore. "Theory of quasi-groups." *American Journal of Mathematics*, **59**(4):983–1004, 1937.

[Hof95] Douglas R Hofstadter. *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought.* Basic books, 1995.

[HQX18] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. "Cooperative Holistic Scene Understanding: Unifying 3D Object, Layout and Camera Pose Estimation." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[HQZ18] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. "Holistic 3D Scene Parsing and Reconstruction from a Single RGB Image." In *European Conference on Computer Vision (ECCV)*, 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." *Neural computation*, 1997.

[HSB19] Felix Hill, Adam Santoro, David GT Barrett, Ari S Morcos, and Timothy Lillicrap. "Learning to Make Analogies by Contrasting Abstract Relational Structure." *arXiv:1902.00120*, 2019.

[Hum12] James E Humphreys. *Introduction to Lie algebras and representation theory*, volume 9. Springer Science & Business Media, 2012.

[Hun74] Earl Hunt. *Quote the Raven? Nevermore.* Lawrence Erlbaum, 1974.

[HW17] Dokhyam Hoshen and Michael Werman. "IQ of Neural Networks." *arXiv preprint arXiv:1710.01692*, 2017.

[HXZ19] De-An Huang, Danfei Xu, Yuke Zhu, Animesh Garg, Silvio Savarese, Li Fei-Fei, and Juan Carlos Niebles. "Continuous Relaxation of Symbolic Planner for One-Shot Imitation Learning." In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[IPS11] Véronique Izard, Pierre Pica, Elizabeth S Spelke, and Stanislas Dehaene. "Flexible intuitions of Euclidean geometry in an Amazonian indigene group." *Proceedings of the National Academy of Sciences (PNAS)*, **108**(24):9782–9787, 2011.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *International Conference on Machine Learning (ICML)*, 2015.

[Jam91] William James. *The Principles of Psychology.* Henry Holt and Company, 1891.

[JBJ08] Susanne M Jaeggi, Martin Buschkuehl, John Jonides, and Walter J Perrig. "Improving fluid intelligence with training on working memory." *Proceedings of the National Academy of Sciences*, **105**(19):6829–6833, 2008.

[JGP16] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax." *arXiv:1611.01144*, 2016.

[JHM17a] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[JHM17b] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross B Girshick. "Inferring and Executing Programs for Visual Reasoning." In *International Conference on Computer Vision (ICCV)*, 2017.

[JR92] Michael I Jordan and David E Rumelhart. "Forward models: Supervised learning with a distal teacher." *Cognitive Science*, **16**(3):307–354, 1992.

[JSY17] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. "Tgif-qa: Toward spatio-temporal reasoning in visual question answering." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *International Conference on Learning Representations (ICLR)*, 2014.

[KK79] Gaetano Kanizsa and Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*, volume 49. Praeger New York, 1979.

[KKL15] George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez. "Symbol acquisition for probabilistic high-level planning." In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[KMG13] Maithilee Kunda, Keith McGreggor, and Ashok K Goel. "A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations." *Cognitive Systems Research*, **22**:47–66, 2013.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[KSM17] Ken Kansky, Tom Silver, David A Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. "Schema networks: Zero-shot transfer with a generative causal model of intuitive physics." In *International Conference on Machine Learning (ICML)*, 2017.

[KW13] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes." *arXiv:1312.6114*, 2013.

[KW16] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907*, 2016.

[KZS15] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. "Skip-thought vectors." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[Law57] Reed Lawson. "Brightness discrimination performance and secondary reward strength as a function of primary reward amount." *Journal of Comparative and Physiological Psychology*, **50**(1):35, 1957.

[LB14] Matthew M Loper and Michael J Black. "OpenDR: An approximate differentiable renderer." In *European Conference on Computer Vision (ECCV)*, 2014.

[LB18] Brenden Lake and Marco Baroni. "Generalization without Systematicity: On the Compositional Skills of Sequence-to-Sequence Recurrent Networks." In *International Conference on Machine Learning (ICML)*, 2018.

[LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, **86**(11):2278–2324, 1998.

[LF17] Andrew Lovett and Kenneth Forbus. "Modeling visual problem solving as analogical reasoning." *Psychological Review*, **124**(1):60, 2017.

[LFU10] Andrew Lovett, Kenneth Forbus, and Jeffrey Usher. "A structure-mapping model of Raven's Progressive Matrices." In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2010.

[LHH20] Qing Li, Siyuan Huang, Yining Hong, Yixin Chen, Ying Nian Wu, and Song-Chun Zhu. "Closed Loop Neural-Symbolic Learning via Integrating Neural Perception, Grammar Parsing, and Symbolic Reasoning." In *International Conference on Machine Learning (ICML)*, 2020.

[Lin91] Jianhua Lin. "Divergence measures based on the Shannon entropy." *IEEE Transactions on Information theory*, **37**(1):145–151, 1991.

[LLG12] Daniel R Little, Stephan Lewandowsky, and Thomas L Griffiths. "A Bayesian model of rule induction in Raven's Progressive Matrices." In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2012.

[LMM01] Richard Le Grand, Catherine J Mondloch, Daphne Maurer, and Henry P

Brent. “Neuroperception: Early visual experience and face processing.” *Nature*, **410**(6831):890, 2001.

[LTF09] Andrew Lovett, Emmett Tomai, Kenneth Forbus, and Jeffrey Usher. “Solving geometric analogy problems through two-stage analogical mapping.” *Cognitive science*, **33**(7):1192–1231, 2009.

[LWP09] Liang Lin, Tianfu Wu, Jake Porway, and Zijian Xu. “A stochastic graph grammar for compositional object representation and recognition.” *Pattern Recognition*, **42**(7):1297–1307, 2009.

[Mad88] Penelope Maddy. “Believing the axioms. I.” *The Journal of Symbolic Logic*, **53**(2):481–511, 1988.

[Mar82] David Marr. *Vision: A computational investigation into.* WH Freeman, 1982.

[Mar98] Gary F Marcus. “Rethinking eliminative connectionism.” *Cognitive psychology*, **37**(3):243–282, 1998.

[Mar01] Gary Marcus. *The algebraic mind.* Cambridge, MA: MIT Press, 2001.

[Mar18] Gary F Marcus. *The algebraic mind: Integrating connectionism and cognitive science.* MIT press, 2018.

[Mar20] Gary Marcus. “The next decade in AI: four steps towards robust artificial intelligence.” *arXiv preprint arXiv:2002.06177*, 2020.

[McC60] John McCarthy. *Programs with common sense.* RLE and MIT computation center, 1960.

[MD19] Gary Marcus and Ernest Davis. *Rebooting AI: building artificial intelligence we can trust.* Pantheon, 2019.

[MD20] Gary Marcus and Ernest Davis. "Insights for AI from the human mind." *Communications of the ACM*, **64**(1):38–41, 2020.

[MDK18] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. "Deepproblog: Neural probabilistic logic programming." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[Mey51] Donald R Meyer. "The effects of differential rewards on discrimination reversal learning by monkeys." *Journal of Experimental Psychology*, **41**(4):268, 1951.

[MG14] Keith McGreggor and Ashok K Goel. "Confident Reasoning on Raven's Progressive Matrices Tests." In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 380–386, 2014.

[MGK19] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision." In *International Conference on Learning Representations (ICLR)*, 2019.

[Mit93] Melanie Mitchell. *Analogy-making as perception: A computer model*. MIT Press, 1993.

[MKG14] Keith McGreggor, Maithilee Kunda, and Ashok Goel. "Fractals and ravens." *Artificial Intelligence*, **215**:1–23, 2014.

[MMT16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. "The concrete distribution: A continuous relaxation of discrete random variables." *arXiv:1611.00712*, 2016.

[MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[MSD18] Can Serif Mekik, Ron Sun, and David Yun Dai. "Similarity-Based Reasoning, Raven's Matrices, and General Intelligence." In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1576–1582, 2018.

[MTS18] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. "Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[MVR99] Gary F Marcus, Sugumaran Vijayan, S Bandi Rao, and Peter M Vishton. "Rule learning by seven-month-old infants." *Science*, **283**(5398):77–80, 1999.

[New73] Allen Newell. "You can't play 20 questions with nature and win: Projective comments on the papers of this symposium." In William G Chase, editor, *Visual Information Processing: Proceedings of the Eighth Annual Carnegie Symposium on Cognition*. Academic Press, 1973.

[New80] Allen Newell. "Physical symbol systems." *Cognitive Science*, **4**(2):135–183, 1980.

[NH10] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines." In *International Conference on Machine Learning (ICML)*, 2010.

[Pea89] Giuseppe Peano. *Arithmetices principia: Nova methodo exposita*. Fratres Bocca, 1889.

[PGC17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch." In *NIPS-W*, 2017.

[PSD18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron

Courville. “FiLM: Visual Reasoning with a General Conditioning Layer.” In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation.” In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[PZ15] Seyoung Park and Song-Chun Zhu. “Attributed grammars for joint estimation of human attributes, part and pose.” In *International Conference on Computer Vision (ICCV)*, 2015.

[QJH20] Siyuan Qi, Baoxiong Jia, Siyuan Huang, Ping Wei, and Song-Chun Zhu. “A generalized earley parser for human activity parsing and prediction.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[QJZ18] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. “Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction.” In *International Conference on Machine Learning (ICML)*, 2018.

[Rav36] James C Raven. *“Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive.”*. Master’s thesis, University of London, 1936.

[Rav38] J. C. et al. Raven. “Raven’s progressive matrices.” *Western Psychological Services*, 1938.

[RC98] John C Raven and John Hugh Court. *Raven’s progressive matrices and vocabulary scales.* Oxford pyschologists Press, 1998.

[RDG16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You only look once: Unified, real-time object detection.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[RGJ21] Nasim Rahaman, Muhammad Waleed Gondal, Shruti Joshi, Peter Gehler, Yoshua Bengio, Francesco Locatello, and Bernhard Schölkopf. "Dynamic inference with neural interpreters." *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RKZ15] Mengye Ren, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[RR17] Tim Rocktäschel and Sebastian Riedel. "End-to-end differentiable proving." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[SB98] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998.

[SCO11] Daniel L Schwartz, Catherine C Chase, Marily A Oppezzo, and Doris B Chin. "Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer." *Journal of Educational Psychology*, **103**(4):759, 2011.

[SCS13] Claes Strannegård, Simone Cirillo, and Victor Ström. "An anthropomorphic method for progressive matrix problems." *Cognitive Systems Research*, **22**:35–46, 2013.

[SE05] Noah A Smith and Jason Eisner. "Contrastive estimation: Training log-linear models on unlabeled data." In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

[SEM20] Steven Spratley, Krista Ehinger, and Tim Miller. "A Closer Look at Generalisation in RAVEN." In *European Conference on Computer Vision (ECCV)*, 2020.

[SG14] Linsey Smith and Dedre Gentner. "The role of difference-detection in learning contrastive categories." In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2014.

[SG16] Luciano Serafini and Artur d'Avila Garcez. "Logic tensor networks: Deep learning and logical reasoning from data and knowledge." *arXiv preprint arXiv:1606.04422*, 2016.

[SG18a] Snejana Shegheva and Ashok Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SG18b] Snejana Shegheva and Ashok K. Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SH56] Allan M Schrier and Harry F Harlow. "Effect of amount of incentive on discrimination learning by monkeys." *Journal of comparative and physiological psychology*, **49**(2):117, 1956.

[SHB18] Adam Santoro, Felix Hill, David Barrett, Ari Morcos, and Timothy Lillicrap. "Measuring abstract reasoning in neural networks." In *International Conference on Machine Learning (ICML)*, 2018.

[SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, 2014.

[SLV18] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. "Improving generalization for abstract reasoning tasks using disentangled feature representations." *arXiv preprint arXiv:1811.04784*, 2018.

[SM04] Daniel L Schwartz and Taylor Martin. "Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction." *Cognition and Instruction*, **22**(2):129–184, 2004.

[Spe23] Charles Spearman. *The nature of "intelligence" and the principles of cognition.* Macmillan, 1923.

[Spe27] Charles Spearman. *The abilities of man.* Macmillan, 1927.

[SRB17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. "A simple neural network module for relational reasoning." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[SV78] Robert M Shapley and Jonathan D Victor. "The effect of contrast on the transfer properties of cat retinal ganglion cells." *The Journal of physiology*, **285**(1):275–298, 1978.

[TSM15] Kai Sheng Tai, Richard Socher, and Christopher D Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks." In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[TT41] Louis Leon Thurstone and Thelma Gwinn Thurstone. "Factorial studies of intelligence." *Psychometric monographs*, 1941.

[WDG20] Yuhuai Wu, Honghua Dong, Roger Grosse, and Jimmy Ba. "The Scattering Compositional Learner: Discovering Objects, Attributes, Relationships in Analogical Reasoning." *arXiv preprint arXiv:2007.04212*, 2020.

[WG15] Xiaolong Wang and Abhinav Gupta. "Unsupervised learning of visual representations using videos." In *International Conference on Computer Vision (ICCV)*, 2015.

[Wil92] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." *Machine learning*, **8**(3-4):229–256, 1992.

[Win71] Terry Winograd. "Procedures as a representation for data in a computer program for understanding natural language." Technical report, MIT. Cent. Space Res., 1971.

[Wit53] Ludwig Wittgenstein. *Philosophical investigations. Philosophische Untersuchungen.* Macmillan, 1953.

[WJL20] Duo Wang, Mateja Jamnik, and Pietro Lio. "Abstract Diagrammatic Reasoning with Multiplex Graph Networks." In *International Conference on Learning Representations (ICLR)*, 2020.

[WS09] Kilian Q Weinberger and Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification." *Journal of Machine Learning Research*, **10**(Feb):207–244, 2009.

[WS15] Ke Wang and Zhendong Su. "Automatic Generation of Raven's Progressive Matrices." In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[WSG10] Ying Nian Wu, Zhangzhang Si, Haifeng Gong, and Song-Chun Zhu. "Learning active basis model for object detection and recognition." *International Journal of Computer Vision (IJCV)*, **90**(2):198–235, 2010.

[WTK17] Jiajun Wu, Joshua B Tenenbaum, and Pushmeet Kohli. "Neural scene derendering." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[WWX17] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. “Marrnet: 3d shape reconstruction via 2.5 d sketches.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[WXL18] Ying Nian Wu, Jianwen Xie, Yang Lu, and Song-Chun Zhu. “Sparse and deep generalizations of the FRAME model.” *Annals of Mathematical Sciences and Applications*, **3**(1):211–254, 2018.

[WXZ07] Tian-Fu Wu, Gui-Song Xia, and Song-Chun Zhu. “Compositional boosting for computing hierarchical image structures.” In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[XCW15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. “Convolutional LSTM network: A machine learning approach for precipitation nowcasting.” In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[XJZ22] Manjie Xu, Guangyuan Jiang, Chi Zhang, Song-Chun Zhu, and Yixin Zhu. “EST: Evaluating Scientific Thinking in Artificial Agents.” *arXiv preprint arXiv:2206.09203*, 2022.

[XLZ16] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. “A theory of generative convnet.” In *International Conference on Machine Learning (ICML)*, 2016.

[XMY21] Sirui Xie, Xiaojian Ma, Peiyu Yu, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. “HALMA: Humanlike Abstraction Learning Meets Affordance in Rapid Problem Solving.” *arXiv preprint arXiv:2102.11344*, 2021.

[XZW19] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. “Learning Energy-based Spatial-Temporal Generative ConvNets for Dynamic Patterns.” *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

[YGL20] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua Tenenbaum. "CLEVRER: Collision Events for Video Representation and Reasoning." In *International Conference on Learning Representations (ICLR)*, 2020.

[YK06] Alan Yuille and Daniel Kersten. "Vision as Bayesian inference: analysis by synthesis?" *Trends in cognitive sciences*, 2006.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. "Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding." *arXiv preprint arXiv:1810.02338*, 2018.

[ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. "Visual7w: Grounded question answering in images." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[ZGF20] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. "Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense." *Engineering*, **6**(3):310–345, 2020.

[ZGJ19] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. "RAVEN: A Dataset for Relational and Analogical Visual rEasoNing." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZJE21] Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. "Acre: Abstract causal reasoning beyond covariation." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[ZJG19] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. "Learning Perceptual Inference by Contrasting." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[ZJZ21] Chi Zhang, Baoxiong Jia, Song-Chun Zhu, and Yixin Zhu. "Abstract spatial-temporal reasoning via probabilistic abduction and execution." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[ZM07] Song-Chun Zhu, David Mumford, et al. "A stochastic grammar of images." *Foundations and Trends® in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford. "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling." *International Journal of Computer Vision*, **27**(2):107–126, 1998.

[ZWZ16] Jun Zhu, Tianfu Wu, Song-Chun Zhu, Xiaokang Yang, and Wenjun Zhang. "A reconfigurable tangram model for scene representation and categorization." *IEEE Transactions on Image Processing*, **25**(1):150–166, 2016.

[ZZH17] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. "Structured attentions for visual question answering." In *International Conference on Computer Vision (ICCV)*, 2017.

[ZZW19] Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. "Abstract Reasoning with Distracting Features." In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[ZZZ19] Chi Zhang, Yixin Zhu, and Song-Chun Zhu. "MetaStyle: Three-Way Trade-off among Speed, Flexibility, and Quality in Neural Style Transfer." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[ZZZ20] Wenhe Zhang, Chi Zhang, Yixin Zhu, and Song-Chun Zhu. "Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.