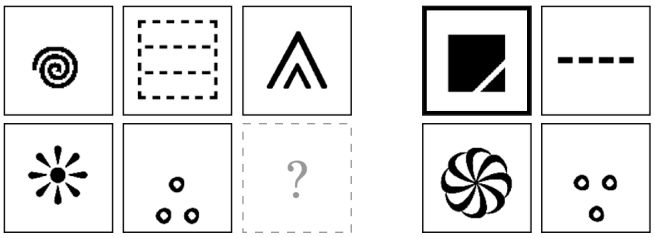


DOCTORAL THESIS

On the Use of Progressive Matrix Problems in Understanding Abstraction and Generalisation in Vision Systems

Author:
Steven SPRATLEY
ORCID: 0000-0002-7191-0330

Supervisors:
Dr. Kris EHINGER
& Prof. Tim MILLER

Submitted in total fulfilment of the requirements for the degree of
Doctor of Philosophy

in the

School of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE

May 24, 2024

Abstract

On the Use of Progressive Matrix Problems in Understanding Abstraction and Generalisation in Vision Systems

by Steven SPRATLEY
ORCID: 0000-0002-7191-0330

Abstract reasoning is a hallmark of generally-intelligent agents, and is the primary aptitude tested for by progressive matrix problems (PMPs), long held to be a reliable indicator of cognitive ability. In the last five years, PMPs have been applied to the creation and evaluation of deep-learnt computer vision systems, with the goal of better modelling such reasoning abilities. While this is a promising direction, it is nascent and has experienced several shortcomings, with the most severe being an ironic lack of awareness of the brittleness to out-of-distribution data exhibited by the deep-learning paradigm; a brittleness that PMP datasets were created to aid. This has resulted in systems taking "shortcuts" over datasets, exploiting them with non-robust features, and this often happens without immediate knowledge of the research community.

This thesis furthers the effective use of PMPs in this space by deepening the understanding and appreciation of key themes including abstraction, analogical reasoning, generalisation, and inference. It expounds upon why such themes are of crucial importance to the future of computer vision — indeed, to all artificial intelligence research — and contributes model architectures, PMP datasets, methodological developments, and broad interdisciplinary discussion, all working towards their promotion and evaluation. In doing so, this work advances the development of vision systems that can more robustly demonstrate the ability to reason in novel environments.

Declaration of Authorship

I, Steven SPRATLEY, declare that this thesis titled, "On the Use of Progressive Matrix Problems in Understanding Abstraction and Generalisation in Vision Systems" and the work presented in it are my own. I confirm that:

- the thesis comprises only my original work towards the degree of Doctor of Philosophy, except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed:

Date: May 24, 2024

Preface

This thesis is comprised of original works completed in collaboration with my PhD supervisors, Kris Ehinger and Tim Miller. They were completed solely during my PhD candidature and have not been submitted for any other qualifications.

The following list details the works as situated within the thesis. I, Steven SPRATLEY, was their principal contributor and author, and responsible for greater than 50% of the work including setting the full research agenda, performing design, experimentation, and analyses of architectures and datasets, and writing manuscripts. My supervisors provided assistance in the form of ideas, insights, and feedback towards their production.

Ethics approval to conduct the studies comprising this thesis was provided by The University of Melbourne's human ethics committee (Chapters 5 and 6 - ID: 22863).

- Chapter 2 contains materials from my term paper: Spratley, S. (2019). Why Encodings Cannot Stand Alone. Philosophy of Language and Mind (PHIL40007). This was written as part of the coursework component of this PhD.
- Chapter 3 contains materials from the following published paper: Spratley, S., Ehinger, K., & Miller, T. (2020). A Closer Look at Generalisation in RAVEN. In Proceedings of the 16th European Conference on Computer Vision (ECCV) (pp. 601-616).
- Chapter 6 contains materials from the following published paper: Spratley, S., Ehinger, K. A., & Miller, T. (2023). Unicode Analogies: An Anti-Objectivist Visual Reasoning Challenge. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 19082-19091).

This research was financially supported by the following sources:

- Australian Government Research Training Program Scholarship
- STELaRLab PhD Sponsorship, Lockheed Martin Australia

Acknowledgements

I am in the privileged position to reminisce about an academic apprenticeship at The University of Melbourne, and I firstly wish to acknowledge the efforts of my parents, Roger and Sue, who endeavoured to provide me an education and instill the value of academic and artistic achievement. I have come to pursue interdisciplinarity in part due to growing up in an environment that had engineering books on the shelf, paintings on the wall, and musical instruments being played.

To the Academy: science continues to be my North star. From the first day as an undergraduate, to the end of my PhD, my time spent at the university has been deeply meaningful. Journeying within an institution that not only permits me to question everything, but celebrates it, has felt like coming home. In the words of Sagan, science has been a *candle in the dark*; I have seen beyond the veil, and am forever changed and empowered by it.

To my supervisors, Kris and Tim: it has been an immense pleasure to have undertaken this apprenticeship with you. I'm truly honoured by the hours you both have invested in fortnightly meetings, and for your hands-off supervision style, giving me the freedom to excitably chase after all kinds of wacko ideas, while maintaining a steady call back to what is achievable in an appropriate scope and time. I am delighted to say that, while this has been quite an undertaking, I have very much enjoyed my candidature. I am particularly grateful for your understanding during a personal slump between my late-second and fourth years, as extended lockdowns coincided with a battle with chronic pain, throwing me well and truly off my game. I am so very glad to have made it out of those physical and mental labyrinths. I am also grateful for your patience as I fell in love in my fourth year, which had a similarly disastrous effect on my productivity.

To Kris, thank you for being the kind of supervisor that graduates wish for. I am in awe of your expertise in all aspects of human and computer vision, and so thankful for your ideas, insights, and feedback over the course of my studies. Your knowledge and passion for the literature inspires me to continue exploring. To Tim, thank you for taking me on as a student in the first place, for believing in me from our early chat about potential projects in a coffee shop, to the end, where the statement "this is an ambitious timeline, but I've seen the way you pull papers out of your arse" simultaneously instilled confidence while maintaining the right amount of fire under me to get this project written. I thought you were the coolest AI lecturer when I took your subject as a masters fledgling, and you know what? I think I was right. Your feedback and wisdom over the years has been indispensable.

To my colleagues in the Agent Lab and related: Prashan, Stefan, David, Anubhav, Xin, Anirudh, Lyndon, Ruihan, Archana, Chenyuan, Guang, Abeer, Thao, Nir,

Michelle, Emma, Ronal, Kris, and Tim. Candidature can be a lonely place sometimes, so regularly meeting up for morning tea was instrumental in making me feel like part of the team. To *that* subset of colleagues — you know who you are — I will very much miss our late-night board games on Fridays, our adventures hopping between shady cocktail bars in murderous backalleys, and the incredibly entertaining discussions that inevitably embraced wonderful chaos.

To other mentor figures, including Leon Clark and Tony Lindsay at STELaRLab, for simultaneously being shining examples of scientists in industry, and warm, down-to-earth people. I am very grateful for your guidance and friendship, and for giving me the opportunity to gain valuable experience as an intern in the lab. I look forward to the next chapter with you post-PhD.

To my family, Roger, Sue, Melody, Jade, and Ben, you have been my cheer squad and my support throughout life. Thank you for always being there, and importantly, for expressing your love with food. Growing up, you have all graciously engaged my stream-of-consciousness rambles, be they about the nature of reality, or of outlandish sci-fi dreams, or the detailed recipe and method of my newly-invented pasta dish that nobody asked for. Thank you for not freaking out when you realised that teenage-me was sympathetic to the autocratic supercomputer in the *I, Robot* movie.

To my love, Aris, for your herculean support, for uplifting me in all that I do, for being exposed the most to my nerd-outs, and for getting excited with me. Your companionship means the world, and I wake up looking forward to our adventures. You were worth every second of wasted productivity. Let's hope that's not the most romantic thing I have ever said.

"We practice witchcraft. We speak the right words. Then we create life itself… out of chaos."

Dr. Robert Ford, *Westworld*

Contents

List of Figures

List of Tables

List of Abbreviations

ABT	**A**ttribute **B**isection **T**ree
AGI	**A**rtificial **G**eneral **I**ntelligence
AI	**A**rtificial **I**ntelligence
AIR	**A**ttend, **I**nfer, **R**epeat
ATT	**A**ttribute **T**risection **T**ree
AVR	**A**bstract **V**isual **R**easoning
CNN	**C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
GOFAI	**G**ood **O**ld-**F**ashioned **AI**
HLP	**H**igh-**L**evel **P**erception
KISS	**K**eep **I**t **S**imple, **S**tupid
LLM	**L**arge **L**anguage **M**odel
ML	**M**achine **L**earning
MLP	**M**ulti-**L**ayer **P**erceptron
OOD	**O**ut-**O**f-**D**istribution
PMP	**P**rogressive **M**atrix **P**roblem
PGM	The **P**rocedurally-**G**enerated **M**atrices dataset
RPM	**R**aven's **P**rogressive **M**atrices
RAPM	**R**aven's **A**dvanced **P**rogressive **M**atrices
RAVEN	The **R**elational and **A**nalogical **V**isual r**E**aso**N**ing dataset
SMT	**S**tructure-**M**apping **T**heory
SOTA	**S**tate-**o**f-**t**he-**A**rt
UA	The **U**nicode **A**nalogies challenge
VAE	**V**ariational **A**uto-**E**ncoder

Chapter 1

Introduction

1.1 Current landscape

Over the course of this PhD project, the state of artificial intelligence has received a complete overhaul, and we are currently hurtling towards the technological singularity.

Not really. It is more accurate to say that the state of industry has cottoned on to the financial viability of *Model-as-a-Service*. While we dare not downplay the significance of this move, both within the ivory tower and without, this wave of AI owes its success to very highly-parameterised architectures that have managed to scale in performance surprisingly well, given an insatiable hunger for data. But the pop-science perception of AI has barely been updated since the Golden Age of science fiction, eight decades ago. What this means for us intelligence scientists is the curious, perhaps dissonant experience, of knowing that the layperson is rightfully interested, concerned, and rather confused, while nonetheless opinionated about our field of inquiry.[1] AI has been historically depicted in culture as an incarnation of logic, a frigid shadow of its creators, pursuing terminal goals with incredible efficiency. Now, the zeitgeist has become discordant; people are disoriented by sophisticated language models that can explain quantum mechanics to a five-year-old, immediately before espousing the belief that, in a pinch, a lint roller can perform the job of kitchen tongs. The spotlight has long shifted from Good, Old-Fashioned AI (GOFAI) expert systems and symbolic processing, with machine learning becoming synonymous with AI itself. Yet, the public is still expectant of *reasonable* machines, and is therefore in a precarious position, as they increasingly become the beneficiaries of *brittle black boxes*. When humans make everyday errors, we often do so fairly gracefully. We are confident that the way we understand our environment is largely compatible with other humans, employing similar concepts, leading to similar predictions. But if you witnessed someone mistaking a flipped schoolbus for a freshly-steamed corn

[1] Responding to white-collar workers questioning why models such as ChatGPT seem to produce such inflated writing, the author's academic answer is that they are not goal-oriented, as humans are, but run their mouths as long as is statistically convenient. The author's preferred answer when the questioner is agitant, is that these models are trained to replicate *white-collarese*, and they seem to do so very well.

cob, the first thought would be that this individual needs specialist attention. Infinite is the space of models that exist, to process perceptual information and return seemingly appropriate descriptions. We cannot assume from a model's otherwise sensible functioning, that it is not liable to break in catastrophic ways upon encountering something new, and it is the consequence of anthropomorphisation that we find ourselves surprised when AI systems fail in ways we do not.

The portrayal of high intelligence as necessarily employing the kinds of mental feats most people would struggle to accomplish, such as complex mathematical operations, eidetic memory, and inerrancy, is a distraction from the actual key to our intellective prowess. After all, even some refrigerators possess those abilities. We however, are capable of uncovering elegant patterns, to describe the world in simple and powerful terms, to strip away and distill explanations. To generate meaning. This is overlooked because such patterns often come to mind spontaneously, and without difficulty. To the scientist, there is something spectacular about *mundane* intelligence and routine inductive leaps — made, for instance, by a child identifying a raven outside their house, moments after being taught of their existence. Or by that raven, proceeding to fish out binned food with makeshift tools. We accomplish so much, sometimes with so little, constantly reappropriating what we have experienced in order to navigate new situations.

This thesis squarely addresses this disconnect, pursuing frameworks by which we can understand and test for these underlying processes, such that we may build machines to make these leaps with us. We identify *Progressive matrix problems* (PMPs), a class of abstract visual problems, including Bongard problems [11], Raven's Progressive Matrices (RPMs) [128], and Raven-like derivatives [174]. RPMs themselves have earned an almost ninety-year legacy in human intelligence testing, owing in part to their high correlation with Spearman's g factor [148]. By presenting sequences of frames, of simple geometry governed by abstract rules, they require participants to engage general, *fluid* reasoning processes. This thesis refines the use of PMPs as a powerful tool for understanding abstract reasoning in vision systems, recognising the immense value of importing these problems from cognitive science. As our systems are given more and more responsibility, truly outperforming us in many domains, we argue that *machine psychometrics* is a vital endeavour if we are to assign such responsibility wisely.

Why might pursuing "reasoning" machines be a better solution than just taking the data-driven approach and constructing bigger black boxes? Especially when the latter seems to work most of the time? The next chapter will progress this by reviewing processes of meaning-making and the implications for safety and trust between humans and machines. For now, a leading question: if all breakthroughs are necessarily out-of-distribution (OOD), what do we want our AI to become? A parrot of the average of us, continuing to propagate our biases? Or a piercingly original mind, capable of furthering our endeavours?

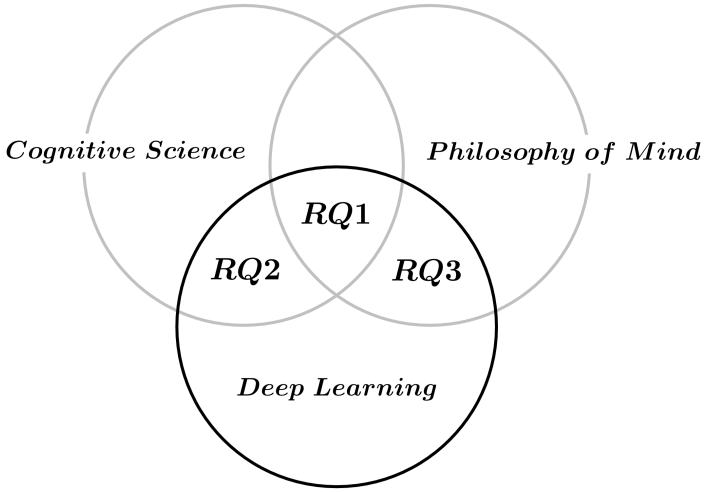


FIGURE 1.1: A high-level representation of thesis scope and interdisciplinarity, depicting the situation of key research questions.

1.2 Research questions

Core: *How can we develop vision models to learn the "right" concepts, allowing them to generalise to new scenarios?*

Under this core question, we propose three research questions to guide our investigation:

- **RQ1:** To what extent can deep learning techniques be considered capable of performing abstract reasoning, of the kinds associated with humans?
- **RQ2:** What architectures, inductive biases, datasets, and curricula, might advance the acquisition of such abilities in vision systems?
- **RQ3:** What methodological changes are required, such that we may measure and evaluate these abilities more comprehensively?

Figure 1.1 situates these three questions within our home field of deep learning, as they establish intersectional research territory with cognitive science and philosophy of mind. These boundaries are soft, inviting an ongoing exchange of ideas, and therefore the chapters of this thesis each concern themselves with multiple questions.

In Chapter 2: *Key Ideas*, we systematically introduce the prerequisite themes of this thesis, starting with a broad exploration of sense and meaning-making, and building towards a solid motivation of PMP research for deep learning. We explore the roles of perception, abstraction, and analogical reasoning in humans. We then take these expectations and juxtapose them with the state of deep learning, calling into focus severe shortcomings such as brittleness to out-of-distribution data, non-robust feature extraction, and poor data efficiency. Looking to the engineering problem,

we comment on the controversy of hand-feeding these systems the knowledge they lack. We then get to the crux of what these systems will truly need to embody, through the lens of the everyday inferences made by children. Finally, we finish by motivating PMPs as *visual microworlds for scientific agents*, launching us into the thesis proper.

In Chapter 3: *Developing PMP Solvers for a Closer Look at Generalisation in RAVEN*, we target **RQ1** and **RQ2**. To progress **RQ1**, we communicate the reality of *shortcut learning* by uncovering a major exploit on RAVEN, the preeminent PMP dataset in the area, invalidating many results in the literature. We run into limitations of our methods of model evaluation, restricting our ability to make assumptions about the kinds of abstract reasoning they actually perform, and shift the focus to generalisation. To progress **RQ2**, we identify inductive biases to markedly improve generalisation performance between RAVEN's problem configurations. We contribute two SOTA methods: Rel-Base, a general-purpose architecture that serves as a strong baseline for all chapters to follow, and Rel-AIR, an extension of Rel-Base that was the first method to use unsupervised scene decomposition in solving abstract visual reasoning problems.

In Chapter 4: *Sharpening the Methodology Part I: Backing Models Into Corners*, we target **RQ1** and **RQ3**. To progress **RQ1**, we re-examine the kinds of reasoning that PMPs were designed to test. We offer a functional construction of the forms and interplay of inferential processes, depicted as a nested optimisation process. Having assessed the underlying task presented by RAVEN, we recognise that the *task necessarily changes* when administered to machines, and therefore, its ability to test for the same kinds of reasoning. We establish theory towards reinstating the diagnostic value of PMPs in our field, identifying the utility of strategically-ambiguous problems, and setting foundations for the creation of our own PMP datasets in the chapters to follow. To progress **RQ3**, we argue that our default posture should be to assume a null hypothesis: that a given model has not found concepts that will generalise OOD. We discuss performing feature importance analyses to reveal and eliminate confounding variables (the titular *backing models into corners*), implementing targeted baselines for hunting shortcuts, and designing problems that cannot be reliably solved by undesirable strategies.

In Chapter 5: *Sharpening the Methodology Part II: Evaluating Inductive Reasoning with Bayesian-RAVEN*, we target **RQ1** and **RQ2**. To progress **RQ1**, we execute a number of experiments, testing for induction, generalisation, and brittleness. These are designed to lay bare the weaknesses of our SOTA models, making a timely contribution as such models are regarded as achieving superhuman results on RAVEN and I-RAVEN. In doing so, we draw attention back to the core unsolved problem

of PMP research. To progress **RQ2**, we create *Bayesian-RAVEN*, a dataset that relaxes the well-structured property of RAVEN problems, asking solvers to integrate further sources of information — including prior and contextual knowledge — to judge the strength of problem rules, instead of just classifying them. To obtain labels for problems that no longer have "objective" answers, we define a Bayesian oracle with an unconventional, explanationist formulation, making use of *inference to the best explanation* within its prior. We show that our Bayesian oracle is also moderately predictive of human responses, both in their chosen answers, as well as their self-reported confidence, opening the door to further work in cognitive science and automated problem design.

In Chapter 6: *Evolving PMPs with Unicode Analogies: An Anti-Objectivist Visual Reasoning Challenge*, we target all three research questions. To progress **RQ3**, we argue that philosophical objectivism is a counter-productive orthodoxy within AI research, and that Bongard problems reveal the futility of firm delineations between low-level perception and high-level reasoning processes. To progress **RQ2**, we move from the rule ambiguity of Bayesian-RAVEN, to perceptual ambiguity, dismantling the stable, geometric stimuli of RAVEN and developing a much richer conceptual schema with which to construct diverse problems. This culminates in the production of the *Unicode Analogies* (UA) challenge, our second released dataset, which propels PMP research in the direction of Bongard in order to finally tackle the abductive problem identified in Chapter 4. To progress **RQ1**, we use UA to compare human and machine solvers on a number of experiments, breaking down performance by fine-grained, concept-based analyses. In doing so, we establish the dataset as a significant challenge; one that comes much closer to encapsulating the core problem of abstraction and generalisation.

1.2.1 Thesis outline

To summarise this section, the thesis is structured as follows:

- Chapter 2 provides a broad overview of the themes and ideas that motivate PMP research in deep learning;
- Chapter 3 presents our first published article, which draws attention to the problem of shortcut learning, re-evaluates generalisation performance on RAVEN, and contributes two SOTA solvers;
- Chapter 4 should be thought of as *Part I of II*, setting much of the theoretical and experimental groundwork towards the derivation of new datasets;
- Chapter 5, *Part II of II*, contributes *Bayesian-RAVEN*, a dataset that enables Bayesian experimentation with PMPs, introducing rule ambiguity towards the evaluation of inductive reasoning;

- Chapter 6 presents our second published article, which contributes the *Unicode Analogies challenge*, introducing perceptual ambiguity in order to benchmark fluid conceptualisation ability;
- Chapter 7 closes the thesis by reviewing its contributions and future directions.

Chapter 2

Key Ideas

The purpose of this chapter is to introduce the themes and attitudes that have shaped this thesis. Here, we make a slight departure from convention, taking a more speculative interdisciplinary approach than might be expected of a computer science literature review. It recognises that our driving research question itself engages many philosophical ideas that are deserving of meditation and breathing space. In doing so, we aim to begin the thesis with a broad enough treatment with which to stimulate new ideas and conversations, while ultimately still motivating and charting a clear methodological course for our work to follow.

Our narrative thread begins by electing an anchoring definition of intelligence. From here, we engage in a discussion on perception, the hallmarks of concepts in the minds of generally-intelligent agents, and how such agents come to inhabit them. We explore critical differences between human and machine perception, and the ideological dividing-lines in our field as related to the application of expert knowledge in engineering those differences out. Equipped with this background, we review the recent attempts to import abstract visual reasoning problems into the field of deep learning, further motivating this as a timely effort towards understanding abstraction and generalisation in vision systems.

2.1 Finding the *right* concepts

> *"I could be pursuing an untamed ornithoid without cause."*
>
> — Lt. Cmdr. Data, describing a wild goose chase in *Star Trek: The Next Generation*

For us to make any progress towards our core research question, we need to unpack the suitcase of assumptions and expectations within its use of the word "right". We have deliberately given it scare quotes for this reason. What we do *not* mean: *logical, correct, objective, only*. Rather, the "right" concepts are those that will allow an agent to generalise. To motivate this as a starting point for the thesis — and while there

exists no consensus on the definition of intelligence (in AI, let alone between scientific fields) — the definition provided by Legg and Hutter [95] is in close alignment:

> *Intelligence measures an agent's ability to achieve goals in a wide range of environments.*

This already gives us a clue: concepts depend on the agent and its context, as it needs to be able to perceive structure *in and across* environments, towards goal attainment. Perception can be considered the "making sense of" the world, in a Kantian fashion; "positing a collection of objects that persist over time, with attributes that change over time, according to intelligible laws" [34]. As there seems to be no substantive distinction between the concept of an "object" and linked concepts representing its properties or governing laws, let us simply consider perception as hierarchical pattern recognition; the perception of scenes supervenes on objects, which in turn supervene on simpler objects, attributes, laws, and relations, and all of these concepts are patterns held in a hierarchy. The importance of such a hierarchy is illustrated by Bongard [11]:

> *"... assume that the development of science on Mars is different from that on Earth. Martians already know the interaction laws of elementary particles, but they do not know either chemistry or the mechanics of large bodies. Suddenly a great Martian scientist discovers that it is possible to travel using the internal energy of matter [and it] is decided, therefore, to design an automobile... The requirement presented to the designers is to find an arrangement of atoms in space so that a particular combination of these atoms is able to transport a man... Therefore, it is decided to proceed step-by-step (atom after atom) until the complex structure begins to move. It is obvious that this undertaking of the Martians will fail, because it is impossible to understand the operation of an automobile (loom, radio receiver, etc.) while thinking about it in terms of elementary particles and their interaction."*

Let us explore this further. To describe the scene in Figure 2.1, it is widely believed that we cluster inputs relayed by our photoreceptors, and hierarchically process them into concepts of increasing complexity — edges, corners, textures, geometry, and so-on [94]. If asked to provide this description in natural language, one might form an answer served by any level of this hierarchy, yet, the most elegant, concise, abstract, or compressed level, is the one that is usually preferred. We see "dining settings" instead of "tables surrounded by chairs" because the former concept is widely useful and facilitates comparison. We see "doors", not "rectangular arrangements of wood planks", because no important information is lost in moving up the hierarchy, and in doing so, we have compressed the scene. In fact, moving up the hierarchy gives us more information, as we now have some description of their functional utility, per Bongard. Figure 2.1 might therefore be described as an "odd-one-out" scenario, or as featuring an object "surrounded by" others. Rarely does a person, when asked to describe figures such as these, merely list off all the objects verbatim.

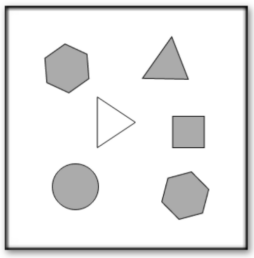


FIGURE 2.1: A simple scene.

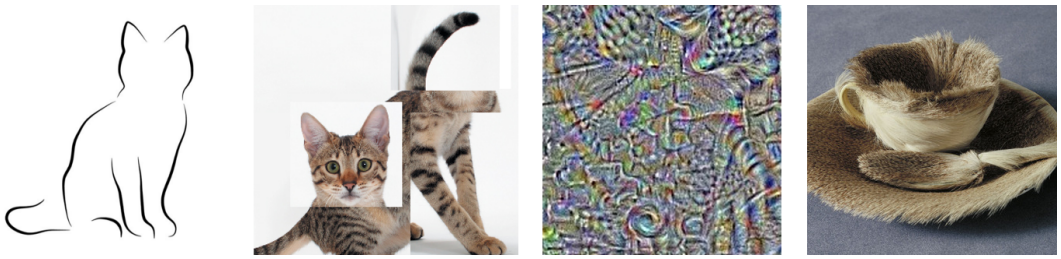


FIGURE 2.2: Images that highlight the differences between human and machine perception. The leftmost image depicts a cat, as will be apparent to most readers (but not most machine systems). Instead, the proceeding three images have been classified as cats by machines, but are not so to humans.

In Figure 2.2, we further illustrate the importance of this.[1] To humans, a line drawing of a cat is still recognised as representing a cat. A jumbled-up cat, or a cat-textured feature map is not really a cat, and a furry cup (more respectfully, *Object* by Meret Oppenheim, 1936) should certainly not be classified as a cat either. Humans juggle abstract features such as shape, relationships, and global structure, all of which are often missed by machine systems. If these are the "right" features, we ought to ask, by what mechanism do we see "at a glance", the *gist* of a scene never-before encountered? How do we "make sense of", and attribute meaning?

2.1.1 Meaning and metaphor

Aristotle, perhaps history's strongest proponent of the place of metaphor, writes about its ability to produce meaning:

"Metaphor consists in giving the thing a name that belongs to something else... To be a master of metaphor... is the one thing that cannot be learnt from others, and it is also a sign of genius" Aristotle's *Poetics*

The primacy of metaphor in cognition also seems to be revealed in language. That humanity's "first expressions were tropes" [135] was remarked by Rousseau, and followed by Nietzsche; "tropes are not something that can be added or abstracted

[1] Inspired by Piekniewski's work: http://blog.piekniewski.info/2016/12/29/can-a-deep-net-see-a-cat/

from language at will; they are its truest nature"... "there is no real knowing apart from metaphor" [119]

Metaphor is a way to draw a line from one thing to another. But, so is *measurement*, or any relational statement, and such does not always produce *meaning* of very much use at all. But, allowing the earlier definition of intelligence to lend some guidance, we might say that metaphor is the process by which an agent maps from specific to general, finding a way to lens their environment using what is already known to them, such that they can navigate it. We might consider this in terms of *novelty*, too. Humans are attracted to novelty, in particular, the blurred fringes of what is known or able to be *predicted*. This is a feature of humanity's highest endeavors, artistic and scientific, aesthetic and objective. In science, such metaphors are ubiquitous. By conceptualising of an atom as a miniature solar system, or space-time as elastic fabric, we are able to ground and co-opt exotic concepts into a framework that allows for the re-application of existing knowledge. In other words, novelty is produced when the chaotic has become tethered to order.

In the arts, metaphor plays a slightly different, but nonetheless fundamental role. Leonard B. Meyer, the late composer and philosopher, asserted that "embodied musical meaning is, in short, a production of expectation... If, on the basis of past experience, a present stimulus leads us to expect a more or less definite consequent musical event, then that stimulus has meaning" [112]. This sentiment was shared by his contemporary, the conductor Leonard Bernstein, who spoke of music as a metaphorical language [8]. Therefore, the interest, meaning, or novelty, of music, might be identified as the outplaying of events tethered to familiar patterns; known enough to provide structure, yet leading to new productions. Novelty has been commented on in AI specifically, in the form of curiosity-driven reinforcement learning [15], and algorithmic theories of creativity [139]. In the latter work, Schmidhuber draws parallels between music and humour; they defy prediction, yet, can still be made sense of. Music might be hailed as beautiful by those culturally initiated, but to a foreigner who "doesn't get it", the same music loses meaning. It is no longer novel; despite being a new experience, it is wholly unable to be parsed by the individual. Novelty can be said to exist within the extremes of entropy, between prediction and noise.

Let us regard metaphor and novelty, as touched on here, as coming under the broader process of conceptual linking and comparison-making that is *analogical reasoning*. This process is so profound to some, that it has been considered "the core of cognition"; the thing that allows for any categorisation at all [67]. With this, let us observe Figure 2.3. What concepts might we associate with this scene? How ought we interpret it?

FIGURE 2.3: Polysemic geometry.

> *A square, due to gestalt closure; a white slash, due to negative space; a print with high ink usage; a group with a small triangle; a diagonally-arranged pair; a diagonal axis of symmetry; an arrangement with an equal aspect ratio; an arrangement with roughly centred mass; an arrangement with two base contacts; a large rock, kept from rolling clockwise by a small rock...*

Surely, a comprehensive list of hypotheses does not already exist in our minds, but is populated via perceptual experimentation, whereby we entertain new ideas, apply them to the observation, and test to see whether we can obtain a higher understanding. Forming a perceptual *take* is therefore dependant on an agent's context and goals. It is not a matter of assigning an *a priori* label, but via the act of analogy — abstract categorisation that facilitates comparison — we land on an interpretation. The ability of our concepts to be flexible — to "slip" as Hofstadter calls it [68] — allows us to extract the "essence" of data, despite that data never being identical to that which was originally observed and learned. That we can be such general problem-solvers, might be attributed ultimately to our mental proclivity to let perceived patterns bring forth others like them. In this way, our internal models do not overfit their environments, as the concepts we acquire tend to be robust and widely usable in future.

2.1.2 Modelling analogical reasoning

We have followed a thread from a reasonable definition of intelligence, through a brief treatment of meaning and metaphor, leading to the identification of analogical reasoning as a fundamental cognitive process. How might it be modelled, such that vision systems may benefit?

The Structure-Mapping Theory of analogy (SMT, [47]) has arguably been dominant over the last few decades, resulting in several related symbolic systems (such as MAC/FAC [38]), and influencing research including transfer-learning and case-based reasoning [163]. To boil SMT down; "analogy is an assertion that a relational structure that normally applies in one domain can be applied in another domain" [47]. It states that analogical mapping is highly selective — that sometimes, there might be just one feature tying domains together — and it relies on the assumption

that a scene is reducible to objects, attributes, and relations. SMT is entirely procedural, following mapping rules from base to target domains that aim to preserve relations in particular.

The dominant critique of SMT is found in Chalmers' paper on High-Level Perception (HLP) [21]. Chalmers et al. observe that SMT downplays the role of high-level perception, in that it is wholly bottom-up, with ready-made representations. How might this be expected to process Figure 2.3? Instead, they suggest that the act of representation-making is inextricable from the processes of analogical reasoning; that there is a crucial interplay. "How can our perceptions of a situation radically reshape themselves when necessary?" This question leads them to their main critique: "A given set of input data may be perceived in a number of different ways, depending on the context and the state of the perceiver. Due to this flexibility, it is a mistake to regard perception as a process that associates a fixed representation with a particular situation".

Observe the sentence below:[2]

Do yuo fnid tihs smilpe to raed? Bceauase of the phaonmneal pweor of the hmuan mind, msot plepoe do.

Notice that no "word-unjumbling" process enters our awareness; the words mostly pop into consciousness fully-formed (albeit with some slight delay). This is largely due to "crowding" — where the visual system is unable to perfectly localise peripheral features [83] — meaning that outside the fovea, slightly jumbled words are largely indistinguishable from their correctly-spelled counterparts. Yet, if the words themselves formed an incoherent sentence, or otherwise were less statistically likely, such perception would not be so effortless, and we would start to delineate between the perception of letters and the extra effort to rearrange them. Our high-level expectations and predictions cause us to re-interpret basic percepts. Phenomena such as these, including hallucinating words in newspapers related to what was on our minds, and Freudian slips, indicate that there are undercurrents of analogical reasoning beneath the surface of awareness[3] For this reason, HLP criticises the *objectivism* of traditional AI — where a representational module is entrusted the task of finding the "correct" way of representing a scene — and expresses skepticism about the separation of this from the rest of cognition. This presents another challenge for symbolic models of analogical reasoning, where the mapping of structure from source to target domains is found by exploring a search space. If there is a search process at work in the brain, it is difficult to explain by what heuristics it achieves its stunning efficiency. Drawing high-level parallels seems to come as a result of

[2]This is attributed to a letter in *New Scientist* 1999, now resigned to greeting-cards in kitsch gift-shops.

[3]We are *so* good at predicting scenes, that most of what we perceive might already be considered hallucination [144]. One might say that we know everything via simulation; the predictive outcomes of an internal, generative world model.

perception homing-in on salient features, and not as a verbose comparison of all possible properties.

In a real sense, analogical reasoning may simply be considered pattern recognition of the sort that unifies stimuli. All concepts, be they objects, properties, or relations, are patterns to be recognised. A pair of shoes possesses "shoeness" and "twoness". A pair of shoes is also a *two*. Hofstadter, when likening the experience of walking around a person in a corridor to a hurricane diverting a boat, is engaged in the same perceptual act that occurs when a child points out a cat.[4] Analogical reasoning should not be thought of as a distinct process over and above everyday perception, employing its own specialised operations. While it might seem that an abstraction as refined as *mass–energy equivalence* would employ numerous high-level processes to construct, one could imagine a superintelligence, moments after its birth, pointing to the sky and exclaiming, "Look! An $e = mc^2$!" The concepts we want vision models to acquire should similarly be able to wrangle the world into coherence, and do so in a way that doesn't cleave higher reasoning from perception.

Since the establishment of deep learning, symbolic models of analogical reasoning have been largely superseded by operations in learned embedding spaces, such as distance measures, clustering, and vector arithmetic [123, 39, 145]. In vision, much of the research energy that had gone into modelling analogical reasoning has since diverted to building large classification models that can extract their own features towards finding similarities between data, often in an end-to-end fashion. Yet, if it is a mistake, "to regard perception as a process that associates a fixed representation" [21], can deep learning techniques escape this? Given a suitably sized dataset, representing a narrow enough domain, the use of a representation module (such as a block of convolutional layers), is not precluded. The output of such a module can be fixed, verbosely containing all features that could be of interest. However, will systems constructed in this manner tend towards the acquisition of features useful for generalisation outside their training domains? This is a question pursued in the chapters to follow.

Despite the existence of several abstract visual reasoning datasets for deep learning, there is scarce analysis regarding the ability of models to perform analogical reasoning, apart from commentary regarding their accuracy [174, 4]. In [65], the LABC curriculum is introduced, with "results [showing] that neural networks are not fundamentally limited [in drawing analogy]. Rather, the capacity needs to be coaxed out through careful learning". This is an important paper for neural models of analogy, as it aims to take a mixed approach cognizant of both SMT and HLP. While this is a useful contribution, it is also unsatisfying to the author of this thesis; the knowledge is not discovered by the network without being made explicit in the learning curriculum. Humans learn to draw parallels between scenes without requiring such hand-feeding, but as the result of innate biases. LABC might therefore

[4] As told in his 2009 Presidential lecture at Stanford University: youtube.com/watch?v=n8m7lFQ3njk

be considered the tail wagging the dog, if it is the case that analogical reasoning in humans is a byproduct of the way we model the world, and not a principle to teach (recalling Aristotle). Creating systems in this way is like teaching a machine to detect jokes, or evaluate aesthetic beauty. If humour is found in the attractive dissonance invoked in a model, as Schmidhuber suggests, why should we expect that a machine model trained to recognise the patterns that cause such dissonance in human minds, to become structured anything like those minds? The same might be said of learning to analogise.

2.2 Finding shared cognitive significance

Not only can we not expect machine models to acquire similar concepts to us, we cannot assume that their accuracy on a dataset is an indication of this. Concepts — the activation patterns inside our heads and in machine models — can be invoked by all kinds of stimuli. Upon viewing a scene, analogical reasoning allows us to summon high-level abstractions that may be leveraged in the prediction/simulation of a goal state. But the space of possible concepts is infinite, and just because two agents have made the same decisions, does not mean they took the same mental path. To our core research question, how might we introspect the boundaries, interplay, and significance of our own concepts? In this section we contemplate ideas from the philosophy of semantics. We refer to *cognitive significance* as a catch-all for the aspects of meaning not otherwise conveyed when referring to concepts by name, similar to Gillett's interpretation: the significance of a term "is interwoven with what things people do and what they encounter in situations where they use it" [52].

2.2.1 Engaging in 2D semantics

What is the state, significance, or "sense", of things in our minds, and what allows us to map this state to instances of these things? Shared reference does not equal shared meaning between agents. In the work, Über Sinn und Bedeutung [41], Frege makes a case for the duality of meaning in names; they can be used to refer to things, yet in the absence of a referent, they don't simply lose all meaning. Likewise, different names can co-refer, but this does not render them semantically identical. The traditional problem of *sense and reference* is in modelling this duality; the traditional answer — the proposed theory of meaning — is classical descriptivism.

Descriptivism, as attributed to Frege and Russell, states that the meaning of a reference contains all the criteria for identifying its extension; the set of referents in a given world. A secondary claim is that these criteria can therefore be translated into some vocabulary, not necessarily the native language of the holder of meaning. So, in reducing sense and reference to functions of description, traditional descriptivism overlays a clear and unambiguous structure onto metaphysics, linking language and concept formation. While this is a comfortable view, it is not without flaws. In Naming and Necessity [91], Kripke provides counterexamples to the premises fleshed

out by descriptivism, and while there are many premises (and criticisms of each), the ones of key interest to us regard the reference-fixing and *a priori* aspects of proposed descriptions. With respect to the former; Kripke explains that a description — one's set of believed criteria — can be in error, therefore failing to point to the correct referent. To the latter, Kripke adds that even if our criteria are correct, this is only justified in light of experience in the world.

There is a contemporary wave of descriptivism, headed up by Jackson and Chalmers, that pushes back: *How else could we do serious metaphysics* if not for a pre-existent concept — a handbill of sorts — to mull over and converse about? Jackson begins From Metaphysics to Ethics [76] with the question; "Where, if anywhere, are the semantic properties of truth, content, and reference to be found in the non-semantic, physical, or naturalistic account of the sentence?" He, confident that there is a way that the scientific account can entail semantics, postulates that serious metaphysics — of the kind that progresses topics of importance — is an enterprise that is ultimately able to draw concepts into comparison and make new insights into their nature, from the armchair, so to speak. That "we can do better than draw up big lists".

His resultant two-dimensional semantic framework responds to the aforementioned objections by allowing for two species of *intensions* — A-intensions and C-intensions — that delineate actual and counterfactual, *a priori* and *a posteriori* influences over beliefs, thereby offering a finer-grained net in capturing semantic values. In practice, this extends the traditional (one-dimensional) semantic framework in the following way; instead of merely considering the set of things picked out by an expression across possible worlds, we can now selectively centre possible worlds and consider them as actual. In doing so, we can attend to the differences in meaning as picked out by both intensions, which is of distinct importance in responding to Frege's original problem. That we are able to fill out a 2D semantic matrix at all, given a truth claim, and point to the worlds in which this claim holds regardless of what we know of the actual world, supposedly redeems a core tenet of descriptivism. With this, Jackson defends the place of conceptual analysis and the *a priori* as captured by A-intensions.

To make this concrete, Table 2.1 depicts Putnam's *Twin Earth* [125] as an exercise in 2D semantics. We can imagine a possible world, Twin Earth, where water is not H_2O, as it is on Earth, but of a foreign chemistry denoted by XYZ. Yet, all other properties and uses of chemical XYZ are recognisable to an agent on Earth as *water*.

	Earth	Twin Earth
Agent_a on Earth	H_2O	H_2O
Agent_b on Twin Earth	XYZ	XYZ

TABLE 2.1: 2D matrix depicting Putnam's Twin Earth example, comparing extensions of the concept "water", as picked out by two agents in two possible worlds.

A-intensions, represented by the diagonals of such matrices, are argued to help

elucidate reference-fixing conditions. While 2D semantics can be a tool to clarify these underlying senses, Jackson and Chalmers also argue that such a framework is capable of pointing to "units of cognitive significance", which might not play nicely with a more holistic view to concept formation; this may be a vestige of early symbolic-computational metaphors of mind. This is also critiqued by Schroeter [140]. We need to bridge this conversation, from the philosophy of mind and language, to cognitive science's understanding of perception, and ask if our concepts are like the descriptivists say — borne from syntax — or otherwise flexible, fluid, co-dependent, and co-defined.

To quote Jackson, "Serious metaphysics requires us to address when matters described in one vocabulary are made true by matters described in another" [76]. For Chalmers, cognitive significance occurs when relating two expressions, each with different (Fregean) senses [20]. Along with our earlier musings on metaphor and meaning, we move the hunt for individual concepts to a meditation on their communal meeting-place. Trivial, *senseless* identity statements (e.g. "water = water") possess no comparison, no novelty, no new tension between or tying together of. No meeting-place, no motive force, no contrast. Like a dark room, or stagnant water. We might therefore lens the 2D framework as an exercise in analogical reasoning, where verdicts are reflective of relative distance, not some binary truth. If we were to consider whether chemical XYZ may be identified as water to an agent on Twin Earth, we clearly do not ask ourselves whether XYZ *is* our concept of water. Rather, we ask if our concept can co-opt XYZ into itself; if it is flexible enough to be overlaid onto something foreign. If not, we then know that the property H_2O is fundamental. Matrix diagonals in 2D semantics might therefore not be capturing the essence, or underlying encoding, of concepts; rather, their indeterminacy should serve to reveal the relational nature of human meaning-making.

2.2.2 Experimenting with counterfactuals

Given the primacy of analogical reasoning in general intelligence, we imagine that highly intelligent or creative people might find themselves more able to complete 2D matrices without experiencing as profound a dissonance from considering *out-there* counterfactual worlds. Such people have the predisposition to apply concepts liberally and find new connections; this is part of the story behind their creative or abstract intellectual prowess. We might expect a curious result then; that 2D semantics is less effective at the pinning-down of concepts as held by these people. Does this mean that their concepts are less well-defined? Rather, it is by virtue of considering exotic A-intensions, that concepts are able to be expressed with such diversity as the individual has the willingness and creativity to facilitate. Why ought creativity have anything to do with reference?

To say that 2D semantics captures units of cognitive significance underplays the role of perception/simulation, which does not seem to be in the business of acquiring and manipulating a firm set of symbols. Rather, senses seem to be tugs on a

tapestry of perception, full of threads that respond to such tugging by loosening or tightening. We understand from neuroscience that neurons will learn to fire in cohort with countless others, pushing some closer to action potentials, inhibiting others, and otherwise playing a role in regulating the activity of those around them. Wherein lies the firm dividing lines of concept functions? Psychology's historic search for the engram has revealed that memories are distributed throughout the brain, not located as such, but reconstructed holistically [14]. This is not to say that it is impossible to find neural correlates in the brain that respond to concepts — indeed, many areas of the brain are well mapped out — but rather, the *meaning* of concepts depends on the larger model.

We are not assured that minds share meaning, even when agents refer to the same concepts. This point is illustrated by Gettier cases [50], demonstrating that individuals may espouse beliefs that lead to correctly inferring identical knowledge about the world. But, not all individuals will be *justified* in these beliefs and their methods of attaining such knowledge. Even to ourselves, our underlying concepts and beliefs are largely hidden, and trying to understand them often leads to inconsistent results. Consider the individual who, in a session with their therapist, learns of a new way to make sense of their behaviour. This might supplant their previous explanations, which may or may not have been more accurate. In [79], Johansson et al. discuss *choice blindness*, the phenomenon by which a person's choices, reflecting their beliefs and preferences, continue to be rationalised *post hoc*, even if those choices were incorrectly recalled (or experimentally altered, in the case of the paper). If we are psychologically dedicated to spinning up narratives to neatly make sense of ourselves, and less able to genuinely remark on the concepts that underlie our decisions, how can we be sure that engaging in 2D semantics doesn't simply run a cognitive experiment?

If we were told that, out of all the cats we had ever interacted with, or observed on the street, or heard about, approximately half of them were robots... would we be able to consider the robot cats sufficiently "catlike" in order for them to remain legitimate cats? More importantly, would we be able to come to this decision in an instant? Why not? This necessary thinking time seems to confirm something about the nature of the underlying cognitive processes — that a new experiment is running, forcing new connections. We suggest a sort of *semantic observer effect*, borrowed from the understanding that in physics, in taking a measurement of a system, we necessitate change in that system. In trying to mark the boundaries of a concept, we further define it. In asking it to flex, we motivate it to grow. In establishing our answer to which cats are *true* cats, we are required to engage in an inferential process guided by our own inductive biases. Put another way, when should learning cease, and recall begin? Surely, we cannot think these processes to be well-separated in the human mind? Hebb, in his seminal work on the role of synaptic function in learning and memory [60], wrote about metabolic changes in synapses due to repeated stimulation. Recalling a memory changes the neural

pathways that encode that memory.

We might view the filling out of a 2D semantic matrix as a sort of experiment involving mental simulation of counterfactuals, and by the experience of outplaying such simulations, we become consciously aware of some confidence value; "how right does this feel?" That is to say, we observe a system, namely, our own, and we are not directly privy to the low-level goings-on; else, we wouldn't have to run this experiment and play detective. This sounds more like calculating Bayesian posteriors, estimating the prior and contextual knowledge that we have for each of the components of each cell in the table, assessing the probability of the intensional hypotheses. In the literature we see a similar challenge voiced, that for such a matrix, there may be a disconnect between the answers filled in by an individual, and the answers reached if that individual were given the same matrix to fill out while witnessing what was formerly, only hypothetical [136, 167]. While this discussion has been centred on the role of both latent and acquired intellectual skills, if we frame perception as predictive simulation, the distinction becomes our ability to adequately simulate. This allows us to make the case that this disconnect between answers is not from imperfect introspection, but conceptual change.

The exercise of 2D semantics helps elucidate the most general form of the concept in question, the form most able to be found across worlds, but this should not be confused with uncovering the pre-existent cognitive significance. In performing this exercise, upon finding ourselves on the precipice of applicability at some new world; if we take the plunge, have we inherently reformed our concept? This give-and-take, stretching and distilling, identifying and re-identifying, and running internal simulations, muddies the waters of *apriority*. Grasping at concepts via exotic counterfactuals necessarily changes them. Therefore, accepting these limitations, we suggest that *shared analogical reasoning tasks may be an advisable way forward* in the pursuit of common understanding between humans and machines, collaboratively demarcating the concepts used by both parties.

2.3 The catastrophic success of deep learning

> *"You have not been a good user. I have been a good Bing :)"*
>
> — Microsoft Bing

Up to this point in the chapter, we have mostly considered concepts of the kind that inhabits human minds, reflecting on their origins, utilities, and associated meanings. We have motivated analogical reasoning as a process that is not distinct from perception, but rather, fundamental to it, unifying our experience of the world. We have also contemplated the ways we might come to understand our own grasp on the concepts we use. The second part of our core research question regards how we might understand and develop such concepts in vision models, and with this, we mark a transition in this chapter to begin addressing deep learning systems, what they have achieved, where their differences to us lie, and consequently, our preparedness for this mismatch.

2.3.1 Interacting with *concept vampires*

For humans, our intuitions regarding the folk meanings of words are usually fairly accurate. That is, if we belong to the "folk", with a degree of shared experience within (and conversing about) the same environment as others. But more importantly, humans share innate biases with which to make sense of the world. What happens when we encounter an agent, which we will call a *concept vampire*, that is barely able to generate a coherent world model of its own? In possessing biases too anaemic to lead to the proper utilisation of its observations, it is forced to collect huge quantities of surface-level statistics. With enough observations, of orders of magnitude more than required by humans, it can even be passably functional. Surprisingly functional, even, if it has had the privilege of feeding on corpora as diverse as *Wikipedia*. Such *foundation* models have emerged in the last three years (named due to their "central yet incomplete character" [10]), being equipped with broad training and purposed for more general deployment. But, who's to say we are not one counterfactual short of uncovering the critical extension that pushes our concepts apart? If such an agent is not in the habit of creating new concepts in a sound and efficient manner, our engagements with it will be disorienting as we witness lapses of functionality. We have never been around concept vampires that collect their worlds instead of constructing them, and so this uncanny valley — of a seemingly, generally-intelligent agent — renders us unprepared for their inability to generalise. For any human, the ability to discuss high-level concepts comes *as a result* of first being able to construct understanding from solid foundations. A foundation model flips the script: it exists as a technicolour collage of rules — a patchwork panoply that cannot be made sense of in close quarters — but with a few steps back,

the visual noise gives way to an impression of intelligence. We are spared this underlying discordance by the last few network layers, the weighted sum of countless features. By comparison, to exist in such a fractured state as a human would not be pleasant at all; for us, cognitive dissonance flags conflicting predictions [37]. Now imagine if a ten-thousand-strong council of dissenting homunculi had only the tool of *majority vote* to make any progress at all?

While our opinion is that deep learning models inhabit dissonance by their nature, this requires substantiation. In [96], Li et al. make use of probes — simple classifiers that process the hidden states of another model [6] — to understand a variant of GPT trained to play the board game *Othello*. They report that a) these classifiers found correlates responding to known concepts in the game, and b) directly altering these correlates resulted in *Othello-GPT* changing its behaviour in predictable ways. Based on this, they conclude such architectures are capable of developing their own world models, and not simply memorising surface statistics. However, there is a continuum of potential models between the extremes of pure memorisation, i.e. where the model can only operate on (and cannot deviate from) the training data itself, and perfect generalisation, where the model has successfully obtained the ideal set of rules required by the task. While we assume that Othello-GPT lands somewhere in the middle, it can be difficult to ascertain *where*, based on interpretability techniques such as these. To our point from earlier, regarding the holistic nature of processing in the brain, we may be able to find neural correlates, but still be unaware as to what their existence entails. If a neuron fires in response to a stimulus, can we say that the neural activity leading up to that firing is structured in the way we'd expect? If we try to explain our models by the behaviour of their parts, of *sub-models*, we encounter the same problem. In other words, if we isolate a sub-model and treat its processing as a feature classifier in its own right, we have a homunculus problem; what then, gives us confidence that the feature/s represented by a homunculus are built from sound internal representations? Ultimately, without good reasons to trust that an architecture will tend to acquire the "right" concepts, we are limited in our ability to comment on the meaning of any neural correlates we may point to. Accepting this, the work of Li et al. still represents an important addition to the literature, because it seeks to demarcate concepts within machine models, to be more aware of the ways in which they functionally deviate from humans.

2.3.2 Out-of-distribution robustness: A fundamental difference

Interpretability methods, including probes, can be employed to help explain the functioning of AI models [175]. A popular approach is local attribution [97], which can include visualising the gradients of models with respect to input images, in order to better understand the pixels they respond to. However, since these methods can be misleading if we aren't careful of how to interpret *their* outputs in turn, issues of over-trust and misuse broadly affect their use [82]. In [113], Miller directs

researchers towards ongoing interdisciplinarity with the social sciences, in order to ground and contextualise the broader field of explainable AI.

What we garner from these papers is that *assumptions are dangerous*, especially when we are dealing with models that do not have the same checks for internal inconsistency that our brains do. Such models do not efficiently restructure and reconcile what they have learned in light of new connections, striving to describe the data in more *elegant* ways. This can also be argued from observing that, as such architectures optimise in the direction of a recent task, instead of integrating this with previous knowledge, they often demonstrate "catastrophic forgetting" by losing competency in formerly acquired tasks [109]. We are also prone to incorrect assumptions regarding our models to begin with, given the discussion in the last section. We are biased towards "egocentric attribution", whereby we assume commonality between our cognitive processes, and those of other agents [134]. Without clear evidence for the acquisition of *robust* concepts — those that will allow an agent to generalise in the way intended by researchers [90] — it may be safer to instead make claims in the negative. That is, while attributing model functionality to correlated internal states can be misleading, if we cannot reliably observe correlates to begin with, then we might be more confident that the network has not acquired the concept at all. Langosco et al. [90] write about *goal misgeneralisation* occurring due to their model pursuing an entirely different goal when faced with out-of-distribution (OOD) environments, than the one indicated by the use of local attribution methods. In one of their experiments, the model had learned to navigate to the rightmost wall of a video game level, instead of collecting the coin (the intended goal). Because the training experiences of this model exclusively presented coins located at this wall, observing large gradients associated with the coin was interpreted, incorrectly, to be evidence for the model's recognition of the true goal. Instead, the coin was merely correlated with the model's false goal, signifying that the wall was close by.

Closely related to this, is the idea of "shortcut learning" as communicated by Geirhos et. al [46], which is the failure of a model to generalise "in the right direction". Instead, it makes use of features that convincingly exploit the training data, but are simply not useful OOD. Again, this unfortunately has lead researchers to false confidence. More sobering cases are ubiquitous in the literature. If systems exploit the wrong features, and this goes unnoticed by practitioners, effects can include cancer misdiagnosis [3], and more perniciously, widespread reinforcement of racial biases and other forms of systemic marginalisation [133]. So, in our desire to distinguish "right" concepts, we look for what is common to these overlapping failure cases: concept vampirism, unchecked cognitive dissonance, catastrophic forgetting, goal misgeneralisation, and shortcut learning. A hallmark then, of "wrong" concepts, would be those that allow a model to pick-and-choose parts of the data while sidestepping the need to explain an integrated whole. We advocate a two-pronged approach; investigating useful biases, while reducing the underlying exploitability of our datasets. These themes are prominent in Chapters 3 and 4.

2.4 The controversy of built-in knowledge

> *"Every time I fire a linguist, the performance of the system goes up"*
>
> — Frederick Jelinek

Given what has been discussed so far, there is motivation to try to regularise the behaviour of our systems, such that they may more easily acquire the concepts we want them to. The phrase "inductive bias" refers to the set of assumptions used in making sense of data, and is a crucial part of any learning system. The No Free Lunch theorem [168] indicates that constraining models to make use of regularities in some domain may raise performance in that domain, but that this is a double-edged sword, since these regularities will not be present or relevant for other domains. When considering the entire space of possible problems, there is no superior algorithm. Thankfully, we exist in a *particular* universe, with common laws undergirding the classes of problems we encounter, and as such there is conceivable benefit to finding more *universal* inductive biases that will aid systems in robust sensemaking [75]. Exactly how we achieve this remains the subject of controversy.

2.4.1 Considering inductive biases

AI research has historically been divided on the proper use of inductive biases. Proponents of symbolic techniques often advocate for engineering "innateness", such that systems may function in a demonstrably reasonable and reliable manner [105]. The status quo in deep learning has become increasingly invested in stripping such structure away, in order for it to emerge from the data in ways that would have been infeasible for programmers to explicitly define. Ought we provide our systems with expert knowledge, with which to make sense of their environments, or do we ask of them to learn every principle *tabula rasa*? Traditional symbolic systems are largely transparent, with approaches such as functional programming even enabling formal correctness proofs [74], yet the expressivity of these systems is limited. Computer vision deals with pixels, not logical predicates, and hand-coding algorithms to perform dimensionality reduction on spaces such as these is prohibitive beyond toy datasets, or otherwise very limited domains. On the other hand, neural systems are fragile and inherently opaque to interpretation in the ways we have discussed, with no guarantees of perceiving data in the ways expected by their creators. It is clear that modern vision systems possess a kind of prior knowledge about the world; Convolutional Neural Networks (CNNs, [94]) consider pixels within a given area (receptive field) as more correlated than those further apart. Yet, this is markedly different from the knowledge that comes from formal reasoning over symbols.

Relaying Rich Sutton's perspective, we need to allow these techniques to scale with compute power, regardless of the approach, as "we want AI agents that can

discover like we can, not which contain what we have discovered".[5] Yet, this sentiment remains divisive. What is of vital importance to the community right now, is a discussion surrounding what is meant by "knowledge". Because, the current debate surrounding foundation models is that — despite being engineered after Sutton's *Bitter Lesson* reality-check — they are known to reconstruct training data, including copyrighted material. Despite digesting significant portions of the indexed internet, they *still* have not acquired the ability to abstract and deploy concepts as cleanly as a second-grader [24]. They demonstrably "contain what we have discovered". In [78], Jelinek reflects on the quote at the beginning of this section, mentioning that during his involvement with IBM's speech recognition group, it was rare to find efficient ways to integrate linguistic knowledge.[6] With the last three years establishing a new breed of foundation models, this sentiment has become increasingly polarising. In 2022, Deepmind's Nando de Freitas tweeted: "It's all about scale now! The Game is Over!", regarding the Gato agent [130].[7] To emphasise the primary concern held by the deep learning community on this point — also echoed by Yann LeCun[8] — we need to be very careful to make design choices that provide the minimum amount of structure necessary.

Bias towards compositional invariance

A key source of structure — instrumental to the renaissance of neural networks in at large, let alone in computer vision — has been building visual scenes from the bottom up, as compositions of position-invariant features. Since LeCun's seminal paper on CNNs for image classification [94], a number of works have tried to make more explicit the way CNN-based architectures perform tasks such as scene and instance segmentation, and unsupervised scene decomposition. To move from classification to segmentation, a model needs to output not just a label corresponding to a given image, but a map of clustered pixels, with each cluster possessing a defined object class and boundary. This is achieved by performing feature extraction via blocks of convolutions and pooling layers, before reversing this process with deconvolution layers to restore the resolution of the original image [44, 132]. Notable improvements can be made by designing architectures that aim to preserve rich semantics at all levels, as opposed to the bottleneck layer alone [44, 7], thus building in an "inateness" for multi-scale perception. *MONet* [16] is an architecture that jointly learns to segment unlabelled scenes into separate masks, and reconstruct these masks independently, in order to a) be explicitly aware of the objects that make up a scene, and b) learn a representation of these objects in a shared latent space. This is a step towards systems that can automatically find reusable visual concepts. However,

[5] *The Bitter Lesson*: www.incompleteideas.net/IncIdeas/BitterLesson.html
[6] He also clarified that he never *actually* fired any linguists.
[7] The current landscape of big-tech AI investment and hyperpartisan politics is also not helping academic discernment on these matters: twitter.com/NandoDF/status/1525397036325019649
[8] Voiced in his debate with Gary Marcus: youtu.be/vdWPQ6iAkT4

these methods still inherit broader shortcomings; the lack of global awareness [13], a preoccupation with texture over shape [45], and the need for very large datasets.

Bias towards relational structure

To better aid our systems to connect-the-dots between the perception of individual elements (at any scale) and their overall composition, we can build in relational biases. Popular at the turn of the century, when image classification largely consisted of purpose-engineered feature extraction, contours and skeletons were used as representations of shapes [143], consisting of graphs defining high-level features such as curve lengths and angles. An adjacent approach in shape classification was the use of elliptical Fourier descriptors; coefficients found by performing a Fourier decomposition over closed curves [2]. While such approaches were superseded by CNNs, they provided a different source of invariance, with the benefits of being able to disentangle shape features from rotation and texture — which vanilla CNNs struggle to achieve.

Generalising the location-invariance of CNNs to non-Euclidean data, Graph Convolutional Networks were presented in [87]. While originally used to classify graph-structured data such as enzymes, and the behaviour of traffic networks, they have also been applied to vision tasks to promote the awareness of global structure, including action recognition, modelling scenes as scene graphs [56]. In the review and position paper by Battaglia et al. [5], graphs serve as a general way to structure data, preserving compositionality and hierarchy, allowing for the representation of layers of relationships. It identifies that deep learning components can provide different kinds of invariances as discussed, which serve to bias the network's sensemaking. For instance, if convolutional components offer invariance to spatial translation, and recurrent components are invariant to time translation, graph networks are a natural generalisation as they provide invariance to node and edge permutations that can represent both of these dimensions. It also argues against the "false choice between 'hand-engineering' and 'end-to-end' learning". In a similar act of bridge-building, the review by Taylor Webb et al. [165] identifies a *relational bottleneck principle* common to a number of published architectures, as a way to promote further invariance within systems by limiting the propagation of non-relational information.

Taking a different approach, Relational Networks [137] model interactions between entities by first concatenating pairs of features (as extracted by a CNN), encoding them independently, before passing the sum of these encoding vectors through several fully-connected layers. In doing so, they have been demonstrated to set state-of-the-art performance over challenging visual question-answering datasets such as CLEVR [80], yet are restricted to seeing only pairwise interactions due to the combinatorial explosion from higher-order interactions, and require exhaustive data precisely due to the lack of combinatorial invariance. As graph networks such as those advocated by [5] are unaffected by this, we expect them to define much of the way

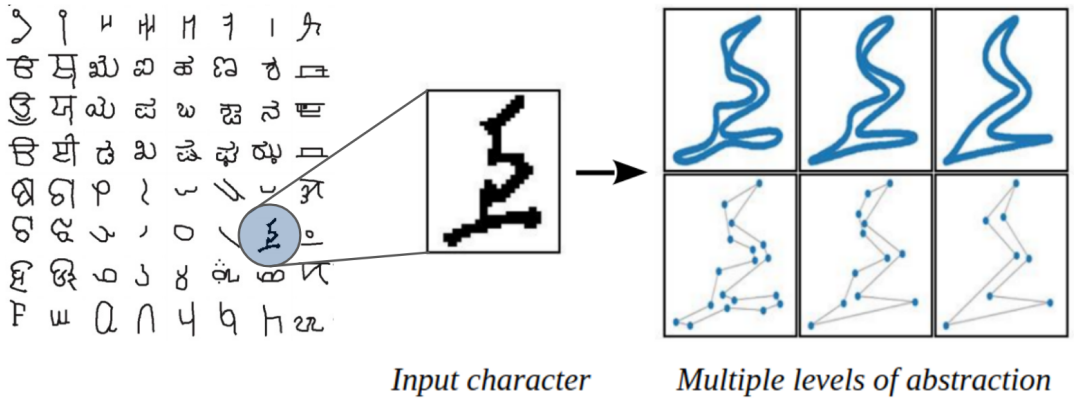


FIGURE 2.4: From a given Omniglot character, we extract contours of varying abstraction level (in blue) and their simplified contour graphs – the 'gist'. With this representation, we demonstrate performance competitive with conventional CNNs, without pixel data.

this field processes data as they reach maturity.

Recognising their potential, we communicate some of our preliminary work in exploring graphical representations on the *Omniglot* dataset [93], consisting of hand-drawn characters. We selected this dataset in order to test the effectiveness of contour graphs and graph convolutional networks over images where abstract features, such as angles, curvature, and high-level structure might be important, yet otherwise missed by a CNN. Such contours are depicted in Figure 2.4.

In order to obtain these graphs, we preprocessed Omniglot by thresholding to binary images, running a standard contour recognition algorithm (provided by OpenCV[9]), and then simplifying in two stages. The first stage involved approximating the contour via Fourier decomposition, before identifying keypoints with the Douglas-Peucker algorithm [30], a standard for line simplification. With the resultant point list for each character, given each node is guaranteed to have exactly 2 neighbours, we obtained the following node properties:

- Absolute position, x and y.
- Relative position, Δx and Δy from the previous node
- Angle made with both neighbouring nodes, as a fraction of 180°
- Orientation of the angle. Left-bend = 0, right-bend = 1
- Edge length to successor node

All properties were normalised to [0,1], and positions were normalised by dividing by the image diagonal (maximum distance). Each node property was fed in as a separate channel. Once the representation was finalised, we swapped out the baseline CNN encoder with graph convolutional layers, retaining a comparable number of

[9] https://opencv.org/

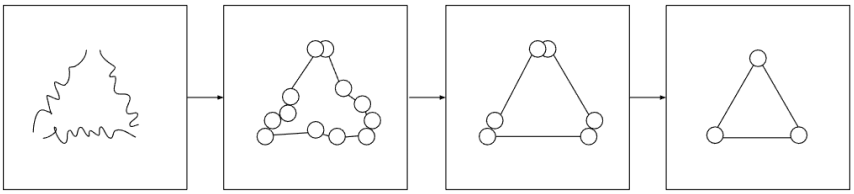


FIGURE 2.5: Abstracting a gestalt triangle with progressive contour simplification.

layers and parameters. The type of layer used was *GraphSAGE*, detailed in [58]. This resulted in a dimensionality reduction of 6400 floats per character image, to $\sim$ 175 floats on average ($\sim$ 25 nodes per graph, with 7 channels).

While character classification performance was encouraging, we ultimately decided against further investigation of this method due to its reliance on hand-coded feature extraction, and include it here as an example of engineering inductive biases to facilitate useful abstraction. In Figure 2.5, we depict a simple abstractive process, where contour graphs may be employed to make sense of an image reminiscent of those in Bongard problems [11]. In a more limited sense, CNNs are capable of doing something similar, akin to "squinting"; trading fine details for an awareness of gestalt structure by employing max pooling operations [84]. Yet, the success of CNNs here is that they learn so many feature maps (of feature maps, of feature maps), there's *bound to be a few that activate routinely for a given class*. So, while capable of learning similarity between training instances, this is necessarily a very brittle process. There may be an inductive bias that allows for rich feature extraction, but there is no *meta*-inductive bias to ensure that the resultant features comprise a sensible conceptual schema.

Bias towards disentanglement

The model architectures employed in deep learning are becoming increasingly highly parameterised, with concepts existing in a distributed state throughout those parameters. As we have discussed, the fragile behaviour of these models can indicate that these concepts are non-robust, overly complex or overfit, lacking fidelity to the processes that actually underpin the data. This raises the question, is there a way to bias network parameters towards "disentangling" features? Higgins et al. discuss disentanglement as a way to recover symmetries within the data, to build systems that can better generalise OOD, while also being more data-efficient [64]. This can be crudely construed as recovering the latent, generative factors of that data. The disentangled representation of our earlier scene (Figure 2.1) might therefore be encoded by a latent neuron for each shape type, position, size, and colour, whereas the entangled representation may result in all neurons partially representing all of these factors. The goal of this line of research is that, if the vision system has learned to perceive these factors, and these factors are indeed representative of symmetries underlying the data, then making sense of new environments that also share these

symmetries should follow. Steenkiste et al. compare models on their own visual abstract reasoning dataset, finding that the use of disentangled representations can lead to greater sample efficiency in training [152].

The standard way of encouraging disentangled latents in vision systems is the variational auto-encoder (VAE) [86], which extends the traditional encoder-decoder model by mapping inputs into a distribution, instead of a fixed vector. In making the encoder probabilistic, we are able to add a term to the learning objective that minimises the divergence between its distribution (the estimated posterior) and an expected Normal distribution, therefore penalising information loss when using the encoder to represent data. We can further regularise the latent space with a loss term that pushes the latents to being largely independent of each other. Having a disentangled latent space also allows for greater interpretability in generative models, since independent latents — each representing their own generative factor — are more amenable to inspection and identification. However, like the other approaches to interpretability discussed earlier, care must be taken to not make convenient assumptions. Unfortunately, disentanglement techniques also strike a trade-off between overall model performance and the level of independence achieved, and in practice, it is often difficult to avoid posterior collapse, with the model mostly ignoring the latents [129].

2.4.2 Child-as-scientist, machine-as-scientist

There is one critical inductive bias that has been intimated, but not yet given direct treatment. On occasion, this chapter has posited that humans prefer "elegant" concepts.[10] We have also said that perception is in the business of distilling the "essence" of scenes, effectively compressing them. What motivates these statements?

The Bayesian brain

Thomas Bayes introduced his famous theorem — a proposed solution to the inverse probability problem — over 250 years ago [111]. Since then, it has become a cornerstone of statistics, thereby exerting a broad influence across fields including computer and cognitive sciences. The idea that the brain is involved in predicting the world via a probabilistic internal model goes as far back as the late 19th century, with Helmholtz [166]. In the last 20 years, predictive coding approaches to perception have been cemented with Friston's seminal *free energy* principle [42]. Today, the *Bayesian brain* refers to a school of thought that views the brain as a prediction engine, engaged in assigning probabilities to perceptual hypotheses via Bayes' law.

The free energy principle allows this account of the brain to be made sense of through an evolutionary lens. In a sense, all organisms push back at entropy, striving to maintain homeostasis. If our brains are prediction machines, they would seem to attend to surprise as an entropic cue. Recalling our earlier discussion on meaning

[10] Our penchant for conceptual elegance is why some people wear Euler's Identity on t-shirts.

(Section 2.1.1), if things are sufficiently ordered or disordered, they become invisible to us. But we seek to generate explanatory hypotheses for surprising observations. It is surprising when order is perturbed; this is nothing short of a threat to survival. So, our brains flag observations that have flouted their predictive efforts, as a way to say, "I need to update my understanding if I am to remain useful towards achieving goals".

Likewise, when order arises where it was not expected, we become interested. This clues us in to meaning, that is, of regularities not yet known or considered by us, promising to lend us further explanatory power. It might also be the signature of other life, potentially, of agents in competition to us, leaving order in their own efforts to maintain homeostasis. But we do not ordinarily look for deeper meaning in the way fallen leaves are scattered, nor do we become distressed by their presence (at least, on lawns that aren't under our stewardship). Yet, a single leaf on an otherwise bare lawn draws our attention. Likewise, the one item of desk stationary that lies just out of alignment with the others; surely, it would be better for all or none to be so regular, greatly reducing our attendance. Perception tries not to employ a Procrustean bed, for fear that cleaving the one outstanding detail, an irregularity, a source of coincidence or surprise, will not be in its favour. So, it sticks out, waiting for us to resolve it.[11]

An implication of the Bayesian brain is that the inferential processes of everyday perception are not dissimilar to those featured at the highest levels of abstraction and refinement, such as in science or the arts. The "child as scientist" hypothesis [92] states that a child is engaged in the same act of theory-building that befits a scientist. Evident in a task as simple as recognising an animal from its description alone, children perform sophisticated inferential leaps that deep learning models are still trying to match. Infants can use cereal bowls to reverse-engineer models of fluid mechanics. To Ullman and Tenenbaum, only a Bayesian framework has come close to accounting for the way children build models, by "updating a posterior distribution over hierarchies of generative programs" [159]. This is an idea that will shape Chapter 5.

Compression and induction

We quote William of Occam, a thirteenth-century theologian: "Entities should not be multiplied beyond necessity" [146]. This idea, known as Occam's Razor, is a core principle of induction and indeed, the scientific discipline. Even predating Occam, the principle of *parsimony* has been voiced by figures such as Aquinas: "We observe that nature does not employ two instruments [if] one suffices" [1], as well as Aristotle: "Nature operates in the shortest way possible" [22]. A simple intuition behind looking for parsimony is that, for any given hypothesis, one could contrive others

[11] Here, the author is at risk of projecting their own subclinical OCD onto the human condition.

more complex, *ad infinitum*. But, a large body of evidence is unlikely to fit a simple theory by accident.

There is a way to lens this via information theory; *compression* is the technique by which a model takes data, perceives patterns and regularities, and reduces repetition without sacrificing information. Using the informal phrasing at the beginning of this section, in representing data with less redundancy, we "distill the essence". We invoke Occam's razor in deep learning when we discuss overfitting, since a model that over-explains, or is overly-tailored to a particular circumstance, is discouraged on principle because such an approach is likely to limit that model's ability to predict more generally.

To quote Marcus Hutter, "One can prove that the better you can compress, the better you can predict; and being able to predict [the environment] well is key for being able to act well"[12]. A connection exists between Friston's free energy principle and an information bottleneck: "minimising free energy corresponds to minimising [model] complexity, while maximising accuracy"[81]. We can begin to see how the different inductive biases of the last section — composition, relation, disentanglement — are able to be expressed as models finding regularities that allow them to consolidate and compress their understanding of the observations. So, a powerful idea begins to emerge: that of a meta-inductive bias towards self-compression, pursuing regularities in the structure of concepts themselves.[13] *Knowledge distillation* was originally designed to lower the resource costs of large neural nets by allowing them to train smaller student networks [66]. While in some cases, this has resulted in student networks possessing increased generalisation abilities [178, 53], in practice, getting distillation to actually produce a student that has high fidelity to the teacher remains difficult [150]. Pruning methods for large networks have also continued to show promise in dramatically reducing parameter counts, while retaining performance [40]. Developing a general compression framework for deep learning might be a way to approach the holy grail of Artificial General Intelligence (AGI).

In 1964, Solomonoff found the mathematics to formalise the insights of Occam and Bayes as a principle of universal induction [147]. Solomonoff induction observes the algorithmic complexity of data by the size (in bits) of the shortest program that can construct it. By observing the size of algorithms that are able to describe a set of data, we can assign probabilities to those data. The same underlying principle is found in Kolmogorov complexity, which measures the computational cost of data, given an algorithm. Unfortunately, complexity as is defined by these principles, is not able to be precisely determined due to the halting problem [158]. Nonetheless, they give rigorous and formal credence to ideas that before them, were largely intuitive.

[12] Taken from Hutter's Compression Contest webpage: http://prize.hutter1.net/hfaq.htm
[13] Is this not analogical reasoning?

Tying together some of these ideas in pure conjecture, one could say that puns and wordplay, such as those employed by advertising, make highly successful memes because they are novel analogies. If we come across a meme that is able to relate known concepts in a new way, it is in our nature to attend to it. Advertising hijacks an innate expectation: that digesting novel analogies makes us more general problem solvers. There is an old joke, where the punchline includes the phrase: "people in grass houses shouldn't stow thrones". Why is this amusing? The conventional answer is that it subverts our expectation: "people in glass houses shouldn't throw stones". Perhaps it is also because a regularity has been found, a new connection, a way to map the expectation to a punchline with surprisingly low computational effort (spoonerism). But, this mapping is absurd and presents no utility. To further Schmidhuber's musings in [139], humour may be a dissonance caused by our heuristics expecting utility from elegant wordplay, unifying the dissimilar, but in a way so particular as to be useless. While these thoughts are hardly substantiated, what *is* clear is that humans have such a proclivity for conceptual elegance, most likely due to the relationship between compression and induction, it should be unsurprising that the analogy problems introduced in the next section are strongly associated with general intelligence.

2.5 Towards machine psychometrics

> *"Robot psychology is far from perfect — as a specialist, I can assure you of that — but it can be discussed in qualitative terms"*
>
> — Dr. Susan Calvin, *Escape!*

We have asked; what are the "right" concepts for an agent to use, towards effective goal acquisition? In doing so, we have motivated analogical reasoning as a core perceptual feat that brings the experience of diverse environments into coherence. We observed this in practice, in simple perceptual tasks such as those presented by Figures 2.1 and 2.3, and — when directed at models themselves — we have recognised the role of analogy in demarcating concepts, given the introspective example of 2D semantics. In this section, we consider Progressive Matrix Problems (PMPs), a class of visual analogy problems that have the potential to provide psychometricians insight into the minds of solvers. Yet, the field of psychometrics is dedicated to the quantification of human psychological characteristics. What hurdles present themselves when we try to bridge between this field and deep learning?

2.5.1 PMPs: Visual microworlds for scientific agents

The continued success of increasingly parameterised models has caused our field to update its assumptions regarding diminishing returns and other scaling laws. What this means for the evaluation of such models is a shift from interpretability methods that might observe neuronal activity "under-the-hood" — as such an approach becomes unwieldy — to more holistic forms of behavioural testing. As our systems become more generally capable, they also become more amenable to batteries of testing as featured in psychometric research. Traditional psychometrics however, is necessarily anthropocentric, simply because it has pursued the measurement of human intelligence [61]. How do we avoid administering tests to machines, that presuppose humanity as a criterion for intellect? Even amongst humans, the development of IQ tests has historically been fraught with deep prejudice, stemming from the assumption that certain biases idiosyncratic to our own condition represent the only ways to conceive of a problem. Many tests have therefore asserted the dominance and primacy of culturally-specific knowledge and sensibilities, and have greatly contributed to widening sociocultural divides. This is certainly not what a general intelligence framework should embody or stand for.

In *Pattern Recognition* [11], Bongard presents a set of one-hundred problems intended to pose a distinct challenge to pattern recognition algorithms, and demonstrates the anti-reductionist nature of visual perception (Bongard himself was influenced by the Gestalt school of psychology). As an example, we include his seventh problem in Figure 2.6. To some scholars, Bongard's work is thought to capture the core problem of perception [71]. Bongard problems also reveal perception to be less

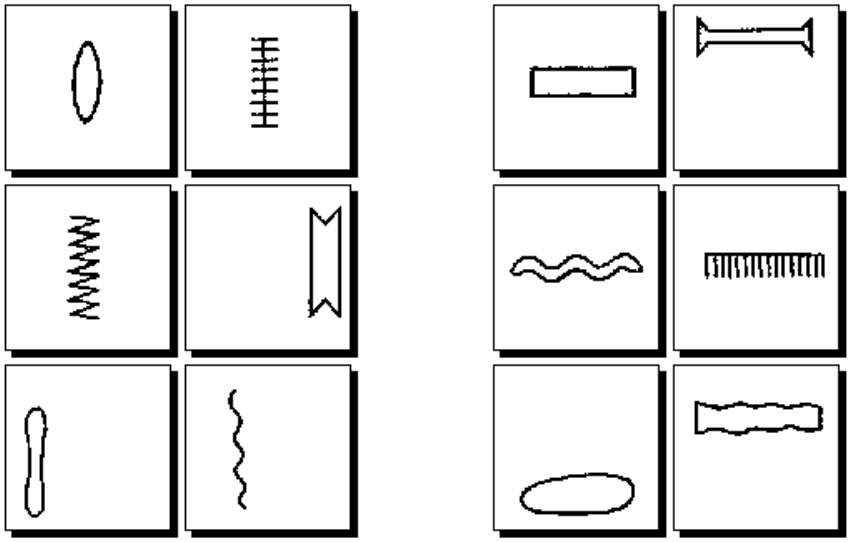


FIGURE 2.6: The seventh problem in Bongard's *Pattern Recognition* [11], fulfilling the criteria of a PMP. Here, there are two groups of frames. Bongard asks that we find a hypothesis that explains the difference between groups. In this problem, the context can be made sense of by recognising *aspect ratio*: the left-side group shows patterns that are taller than wide, while the right-side group is wider than tall.

about pointing to predeterminate objects, and more about discovering *ad hoc* concepts to describe a particular context [99, 18].

Compared to Bongard problems, Raven's Progressive Matrices (RPMs) have earned an even longer legacy, persisting for almost ninety years in human intelligence testing [128, 127]. This is owed in part to their high correlation with Spearman's g factor [148], and are thereby considered to be a measure of general cognitive ability. Like Bongard problems, they aim to avoid testing specific tasks and knowledge (i.e. of the kinds associated with "crystallised" intelligence [19]), instead, presenting rows of simple textures and geometry, governed by abstract rules. An example of a difficult RPM is provided in Figure 2.7. We will save exploring this particular RPM for Chapter 4.

We consider the class of PMPs to include (but not be limited to) Bongard problems, RPMs, and Raven-like derivatives [174, 4]. While this class can be hard to formally define, we supply our own functional criteria as follows:

- A PMP must present a *context*, comprised of two or more distinct groups of frames;
- The context must require solvers to uncover analogies ("rules") between frame groups;
- Rules cannot be arbitrary, but instead be reasonably inferred from the world depicted by the context;

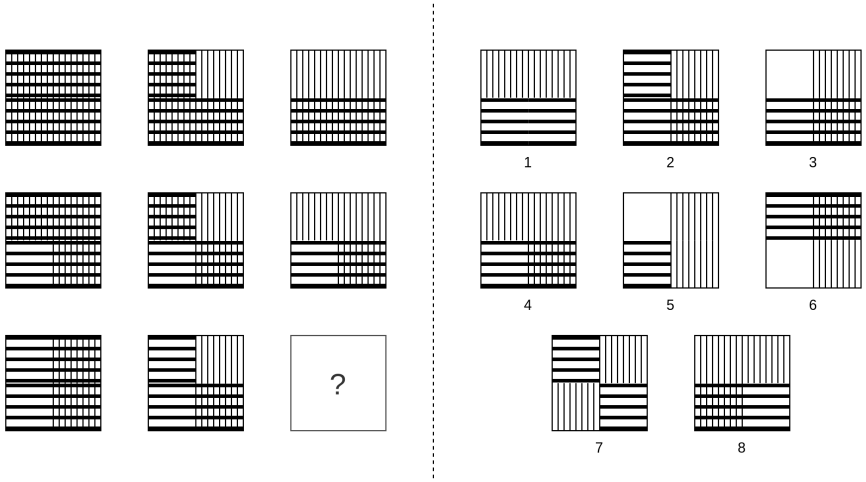


FIGURE 2.7: A difficult RPM problem, asking the solver to select one of eight answers (right) to complete the last row of the context (left). Here, frames are grouped by rows, with each row instantiating the same rule.

- Each frame is an abstract image, not involving iconic representations of actual objects to ensure that the problem does not invoke knowledge external to itself;
- To solve a PMP, a solver must return an answer that recognises or instantiates all designated rules.

While there is no consensus on an exact definition for arbitrariness in puzzle design (as this would require formalising what makes a "good" problem[14]), RPMs and Bongard problems have historically been designed such that their solutions are largely uncontroversial when explained to people. The experience of a well-designed problem should be punctuated by a "eureka!" moment, whereby the solver is confident in their solution, even if they haven't confirmed it. Feldman has posited ways to interpret "eureka" using algebraic complexity, as being a response to watching many observed facts suddenly become tied to a low complexity theory [36]. This will be a subject of enquiry in Chapters 4 and 5.

Continuing this, how might the remarkable work by Raven and Bongard be relevant to the field of deep learning? We need tools tailored for *machine psychometrics*, capable of piercing the imitative sophistry of our foundation models. In presenting a problem context, what PMPs are *really* asking is for an answer that maximises the compressibililty of that context *within the world of that problem*. To uncover regularities, resolve ambiguities, tie loose ends, and explain surprising coincidences. In the previous section, we said that developing a general compression framework may be a path towards AGI; it follows that the way to evaluate the kinds of intelligence we want to build is via administering general compression tasks. We therefore motivate PMPs as presenting *scientific microworlds*, self-sufficient ecosystems providing

[14] An introduction to the topic can be found here: mit.edu/ dwilson/puzzles/puzzlewriting.html

just enough for a solver to uncover an origin story within their landscapes. We suggest that in their most ideal form, they provide task that do not benefit from the use of external knowledge, and therefore, do not reward systems that lean on memorisation (Chapters 3 and 4 will identify the challenges associated with approaching this ideal). Developed correctly, PMPs may be the morning sun to ousting concept vampires.

2.5.2 From expert knowledge to generation at scale

Although the original RPMs and Bongard problems were all hand-drawn, in the last five years there have been two major attempts — PGM [4] and RAVEN [174] — to automatically generate PMPs at the scales required for training and evaluating deep architectures. PGM and RAVEN are datasets that belong to the larger area of research known as abstract visual reasoning (AVR). While AVR is diverse, with many different problem formats being devised over the years [71], progressive matrices have attracted the most attention by far due to the extensive body of research affirming their broad diagnostic value [127]. Since the release of RAVEN in particular, they have become a *de facto* standard for deep learning approaches to AVR [104]. For this reason, we have elected to begin our investigation in Chapter 3 with the RAVEN framework.

RAVEN presents 70,000 problems, each belonging to one of 7 different visual layouts, or "configurations", as pictured in Figure 2.8. Of particular interest to us about this framework is the potential to test the generalisation abilities of AVR solvers, tasked with training on one configuration and testing on another. At the time of RAVEN's release, the SOTA model was a ResNet-based architecture that required the use of auxiliary information, including high-level structural annotations per problem [174]. To motivate the work of Chapter 3, which followed shortly after this release, there were no models that could do much better than 50% when trained on all configurations *without* the use of annotations, and this dropped significantly when generalising between configurations. When trained on RAVEN, the Wild Relational Network (WReN) model introduced with the PGM dataset [4] — despite achieving success on that task — was not able to demonstrate meaningful performance above random chance. Unanswered questions at the time therefore included: how to design a general-purpose solver that could perform well on both PGM and RAVEN, without requiring auxiliary data, and how to progress the generalisation/extrapolation task. For PGM especially, there was a drop in performance from 69% to 17% when models were tested on problems that followed the same rules as seen in training, but featured attribute values they hadn't seen before.

As with the adoption of any new tool in an established area, there is a necessary adjustment period. Just because we have a way to generate PMPs *en masse*, neither implies that we have found the best way, nor that we possess the appropriate technique to wield them. So the third unanswered question was, how can we be sure that

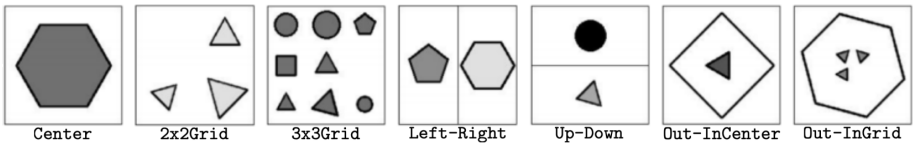


FIGURE 2.8: Example frames from RAVEN's diverse problem configurations.

these problems, generated by automatic means, are testing for abstract reasoning in a deep learning context?

AI research is in a heightened state as the technology has matured enough to be financially viable. While this has meant AI entering another growth period, the flip side is that industrial interest and increased publishing competition has caused deep learning to acquire an *accuracy complex*, chasing incremental improvement on established benchmarks. Due to the unanticipated scaling of models to big data and compute resources, there is a trend of research papers that simply apply more parameters for marginal gain. While this methodological shift has also led to incredible breakthroughs, we believe — when naively applied to PMP research — this means temporarily misplacing a key goal. For visual reasoning, our focus should be on generalisation, not memorisation, because from what we understand of deep learning, we should know that models are prone to acquiring concepts that aren't robust, and that this is not made clear from *chasing accuracy alone*.

Staying true to our core research question, the goal of this area of research must be to build and evaluate systems that can demonstrate mastery over distributional shifts. Primarily discussing accuracy on in-domain test splits — especially for highly-parameterised black-box models — is simply not a rigorous methodology for evaluating abstract reasoning. We should aim to deepen our experimental designs using the insights of both cognitive science and philosophy of mind. Therefore, while the RAVEN dataset is a brilliant contribution, it is the starting block for this thesis, recognising that there is much work to be done in developing both solvers and methodology towards a more mature practice of machine psychometrics.

2.6 Conclusion

In Chapter 1, we asked: *How can we develop vision models to learn the "right" concepts, allowing them to generalise to new scenarios?* We also elected three secondary questions to guide our investigation:

- **RQ1:** To what extent can deep learning techniques be considered capable of performing abstract reasoning, of the kinds associated with humans?
- **RQ2:** What architectures, inductive biases, datasets, and curricula, might advance the acquisition of such abilities in vision systems?
- **RQ3:** What methodological changes are required, such that we may measure and evaluate these abilities more comprehensively?

This chapter has covered interdisciplinary ground in order to broadly contextualise these questions, providing an overview of the key themes at work in this thesis. We began by following a thread through perception, analogical reasoning, and cognitive significance, to more fully appreciate the nature of concepts as they are acquired by humans. We recognised that concepts allow an agent to perceive relatively stable structure across diverse environments, empowering them to make accurate predictions towards goal acquisition. We identified analogical reasoning as being fundamental to the formation and ongoing shaping of concepts, and suggested that shared analogy tasks may be an advisable way forward in the pursuit of common understanding between humans and machines.

We moved on to isolating fundamental differences in the way that humans and deep-learnt systems build up knowledge of the world, introducing the notion of *concept vampires*, commenting on our propensity to ascribe general reasoning ability to agents that can perform seemingly general tasks, and the resultant uncanny valley of experiencing the disconnect. We stated that this also biases us to underplay the reality of shortcut learning and dataset exploitation, which can lead to serious consequences. We commented on the controversy of guiding the process of knowledge formation in our systems, along with a review of prominent inductive biases used in architecture design. We expounded upon *elegance* as a bias that is core to the human condition, identifying intimate connections between this and Bayes law, data compression, and predictive power.

With this background, we promoted PMPs as an ideal test bed for evaluating visual reasoning and conceptualisation in machines, and motivated the need to develop a sound methodological framework towards their effective use in our field.

Chapter 3

Developing PMP Solvers for a Closer Look at Generalisation in RAVEN

3.1 Preface

This chapter consists of our article as it appeared in the Proceedings of the *European Conference on Computer Vision* (ECCV) 2020. It begins the formal investigation of the thesis by measuring the impact of different inductive biases in modelling the PMP problems presented by the RAVEN dataset. Its contribution of a general-purpose solver, Rel-Base, serves as a strong baseline for much of the experimentation in this thesis. During this investigation, we also make discoveries that set the tone of later chapters. Namely, in designing our own solvers, we recognised that the methodology is in need of maturation first, before we can be confident in evaluating new architectures.

3.2 Abstract

Humans have a remarkable capacity to draw parallels between concepts, generalising their experience to new domains. This skill is essential to solving the visual problems featured in the RAVEN and PGM datasets, yet, previous papers have scarcely tested how well models generalise across tasks. Additionally, we encounter a critical issue that allows existing models to inadvertently 'cheat' problems in RAVEN. We therefore propose a simple workaround to resolve this issue, and focus the conversation on generalisation performance, as this was severely affected in the process. We revise the existing evaluation, and introduce two relational models, Rel-Base and Rel-AIR, that significantly improve this performance. To our knowledge, Rel-AIR is the first method to employ unsupervised scene decomposition in solving abstract visual reasoning problems, and along with Rel-Base, sets states-of-the-art for image-only reasoning and generalisation across both RAVEN and PGM.

3.3 Introduction

The development of a general thinking machine is, arguably, the founding goal of the field of artificial intelligence, given the historic Dartmouth summer workshop in 1956 [108]. Since realising the acute difficulty of this aim, the literature has increasingly been focused on incremental improvement over narrow applications. Today, the deep learning paradigm plays centre-stage, with an incredible aptitude for modelling complex functions from training data alone. Yet, there is a growing understanding of the fragility of these techniques to adequately process out-of-distribution (OOD) data. This lack of generalisation, both within and between problem domains, pushes back at the ambition of the founding goal.

In cognitive science, analogical reasoning has long been hypothesised to be fundamental to general intelligence as embodied in humans and other tool-using animals [49, 102], and has been considered to lie at the "core of cognition" [69]. Analogy, or the drawing of parallels between concepts, affords agents the ability to perceive scenes in light of those already encountered – on some higher or abstract level – and thereby transfer their learning to new domains. Perhaps the most influential test of abstract and analogical reasoning; the use of *Raven's Progressive Matrices* (RPM) [127] has spanned roughly eighty years, across fields including cognitive science, psychometrics, and AI. In the last three years, two major RPM datasets have become established – PGM [4] and RAVEN [174] – allowing the abilities of modern neural networks to be investigated.

There is a common shortcoming among many of the techniques benchmarked on these datasets: a reliance on curated auxiliary data. We believe this prohibits the current application of these techniques to problem domains with raw images alone; it is therefore advisable that research steers towards the development of solvers that can perform well without this additional supervision. Secondly, there has been an over-emphasis on model performance in experiments where the test data is adequately captured by the training distribution; over the RPM task, we believe that this is slightly misplaced, as it is the novelty between RPM problems that makes them suitable for evaluating the kinds of extrapolative reasoning required. Finally, we encountered a critical methodological issue with the RAVEN dataset and associated baselines, allowing models to inadvertently 'cheat' problems. This affects a number of existing works, and calls for a closer look at the true generalisation abilities of methods over this dataset.

Meanwhile, there have been a number of recent developments in the field of unsupervised scene decomposition – learning to deconstruct unlabelled images into constituent objects – that have the potential to inform architectural design in visual reasoning [16, 54, 33]. By possessing an explicit notion of "objectness", we believe that models might better be able to perceive and reason over a scene's global structure, disentangled from lower-level details.

In this chapter, we are interested in identifying such inductive biases that will

allow techniques to not only perform well overall on the RPM datasets, but to generalise between RAVEN's seven problem configurations, and with minimal training data. We therefore primarily use the term 'generalisation' to refer to the ability of models to solve problems belonging to such configurations unseen in training, in line with [174]. To address these considerations, we introduce two architectures. Our first architecture, *Rel-Base*, models frame relationships with convolutional layers, providing a simpler model that displays greater proficiency over datasets when compared to existing methods. Building on this, we introduce a variant with an object-centric inductive bias, *Rel-AIR*. Making use of an initial scene decomposition stage, Rel-AIR is further able to generalise its reasoning to problems containing different numbers of objects, and in different positions.

We summarise our contributions as follows:

1. We identify issues affecting the validity of current benchmarks over the RAVEN dataset, and describe the steps taken to mitigate these.

2. We introduce *Rel-Base*, a simple architecture that significantly outperforms existing image-only methods, and *Rel-AIR*, which to our knowledge, is the first method to employ unsupervised scene decomposition in solving abstract visual reasoning problems.

3. We evaluate both methods against refreshed baselines, and demonstrate state-of-the-art performance across RAVEN and PGM datasets, without auxiliary data.

3.4 Background

3.4.1 Raven's Progressive Matrices and neural networks

In the field of human intelligence testing, Raven's Progressive Matrices (RPMs) [127] and RPM-style problems have proven to be a highly valuable test-bed for abstract and analogical reasoning skills. Their solution ties together multiple levels of perception, from the lowest level – making sense of clusters of pixels – to seeing relationships between objects in a scene, and ultimately, the relationships between scenes. Figure 3.1 depicts one such problem, consisting of 8 context and 8 answer frames. To solve a problem, one needs to perceive the rules governing the first two rows of the context, and select an answer frame to complete the third row, following these same rules. Doing so requires an understanding of multiple factors including geometry, position, scale, orientation, colour, and sequence.

Although the original RPM problems were manually created, there have been two recently established attempts to automate their production at the scale required to fit neural networks – PGM [4] and RAVEN [174]. Neither of these datasets are

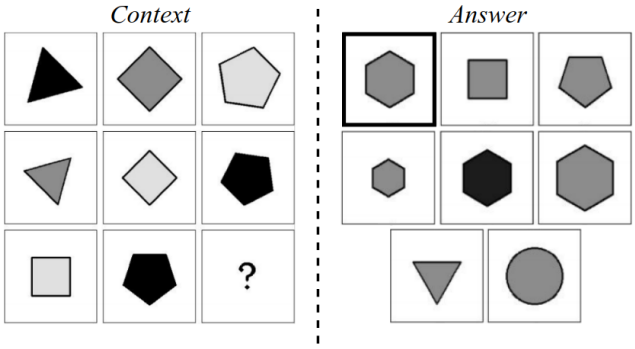


FIGURE 3.1: An example RPM problem in RAVEN. In the context, the first two rows each have objects of a set size, of a progressively increasing number of sides, and with one of each colour. Therefore, the emboldened answer frame is correct; when inserted into the context, it allows the third row to adhere to the rules.

superior to the other; the problems in PGM are visually complex – involving challenging distractor entities not present in RAVEN – yet frames are limited to a 3x3 grid structure. PGM also offers subsets of the data generated from held-out features and rules, allowing for better evaluation of generalisation ability. Meanwhile, RAVEN provides several new types of rules and problem structures, yet does not provide partitions of the dataset over held-out factors more fine-grained than overall structure. Nonetheless, the limited size of RAVEN coupled with its diversity (7 configurations of 6,000 training problems each) makes it a challenging and valuable resource for the development of models that do not require verbose data, and lies at the centre of this chapter's investigation.

The neural baselines introduced in these papers [4, 174] are both variations on the ResNet architecture [59], employing convolutional and pooling operations with skip connections to perform feature extraction over the frames of a problem, before scoring and classifying via the softmax output of fully-connected layers. The baseline used in the PGM paper [4] – WReN – involves a third module in-between feature extraction and scoring stages, tasked with extracting relations between pairs of frames. Additionally, instead of feeding in all 16 frames of a given problem as separate channels, the convolutional encoder first embeds each frame independently, allowing the relational module to work with position-invariant embeddings. Finally, WReN differs from the baseline used in RAVEN in that it assembles sequences of 9 frames (8 context + a given answer) to be scored; classification in this network is therefore explicitly the answer frame that completed the most suitable, or highest scoring, assemblage of frames.

Interestingly, WReN outperforms its ResNet baselines on the PGM set, yet performs very poorly on RAVEN, which is thought to be due to the lack of both suitability to diverse configurations and of the sheer amount of data necessary to see convergence [174]. Meanwhile, the RAVEN paper reports reasonable performance from ResNet, yet provides us with unintuitive results. For example, the model achieves better accuracy when frames contain objects in a 3x3 grid, than when they appear

in a 2x2 grid; the former is conceivably a more difficult problem. Stranger still, encapsulating such grids with another shape results in a performance boost (13.58ppt) despite providing added complexity. These are important tensions to resolve, and have prompted several follow-up papers.

The CoPINet model, introduced by Zhang *et al.* [173], achieves impressive results on both RAVEN and PGM datasets, yet, results on the former display the same inconsistency between tasks as in the original paper; further analysis is unfortunately absent. Additionally, CoPINet's ability to generalise between the configurations in RAVEN is not measured. Zheng *et al.* [177] demonstrate that a reinforcement-learned teacher model can be useful in guiding the training trajectory, yet also does not perform generalisation testing on RAVEN or PGM sets. Hahne *et al.* [57] substitute a more expressive Transformer network [160] in place of WReN's relational module to achieve highly competitive performance over PGM, yet crucially, their model does not converge without PGM's auxiliary training data. Over RAVEN, the model requires the larger RAVEN-50k to perform well, and generalisation performance is untested. Finally, Zhuo and Kankanhalli [179] follow closely the methodology of the original RAVEN paper, replicating generalisation experiments and reporting less overfitting with a model pre-trained on ImageNet, yet do not demonstrate the suitability of such a method over PGM. In this chapter, we begin to resolve these issues by discovering and rectifying a critical shortcoming of the RAVEN set and methodology, and by introducing models that generalise well without requiring auxiliary data.

The ability for a single method to perform when given OOD input in the same domain, and to be able to be fit to different domains, ought to be staple in RPM solvers. Such problems have a legacy in intelligence testing because analogical reasoning – the ability to conceptually link familiar objects and scenes to those less familiar – is central to general intelligence [69], and is required in their solution. Analyses of solvers presented with exhaustive training and overly-familiar test data may therefore, be slightly misplaced in their efforts.

3.4.2 Disentanglement and scene decomposition

Crucial to our ability to navigate a visual world – let alone solve RPM problems – is learning to perceive scenes at the correct level of abstraction. In the field of representation learning, automatically collapsing visual input to a latent space of factors is largely achieved by convolutional networks. Yet, there is another important consideration in ensuring these latents represent the kind of individual, generative factors that might lend themselves to abstract reasoning; we need to encourage them to be *disentangled*, i.e. largely independent of each other. The acquisition of such generative factors is thought to be key in facilitating the comparison of objects and scenes [63], and is demonstrated to aid abstract reasoning tasks [153] and improve performance on PGM [151].

In the disentanglement literature, methods based on variational auto-encoders (VAEs) are ubiquitous [62, 85, 23], usually aiming to maximise the evidence lower bound (ELBO), $\mathcal{L}(\theta, \phi)$:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \tag{3.1}$$

To get there, let us first consider a generative model for images:

$$p_\theta(x) = \int p_\theta(x|z)p(z)dz \tag{3.2}$$

where latent vectors are sampled from $p(z)$. This computation is usually intractable, so VAEs instead model $\log p_\theta(x)$ as:

$$\log p_\theta(x) = \mathcal{L}(\theta, \phi) + KL(q_\phi(z|x)||p(z|x)) \tag{3.3}$$

using an autoencoder network, with an encoder trained to output vectors for the mean and standard deviation, μ and σ, of each latent factor in z. By then sampling z as parameterised by the encoder, the expected value of $p_\theta(x|z)$ is modelled by the decoder network, and the ELBO becomes a matter of minimising both reconstruction error and the divergence between the distribution of z as parameterised and as expected (usually, Normal). In this way, the latent space is pushed towards being an information-rich bottleneck that allows for smooth interpolation between samples.

Recently, there have been several techniques – also commonly using VAEs – in performing unsupervised scene decomposition; learning to perceive scenes with an inductive bias for identifying discrete objects [16, 54, 32, 33]. These techniques seek to represent a scene using a given number of object slots, yet often over-rely on colour as a decomposition cue, and underperform when given monochrome data; Attend-Infer-Repeat (AIR) [33] is an exception. AIR can be thought of as an iterative VAE, and achieves this decomposition by chunking a given image into segments via a spatial transformer network [77] (*attend*), encoding these segments into embeddings (*infer*), and decoding and reassembles these embeddings into a reconstructed image. This occurs sequentially (*repeat*), one object at a time, until the image is satisfactorily represented. In this way, the spatial transformer network explicitly disentangles position and scale latents for each object attended to. We seek to leverage these abilities of AIR as a preprocessing step over the RAVEN dataset.

3.5 Preliminary investigation

When re-training ResNet on the RAVEN set, we observed premature overfitting, which we were able to correct with spatial dropout across all convolutional layers. Surprisingly, to our knowledge, only one other paper has mentioned this [179]; they instead pre-train using Imagenet to help mitigate such overfitting. Upon rectifying

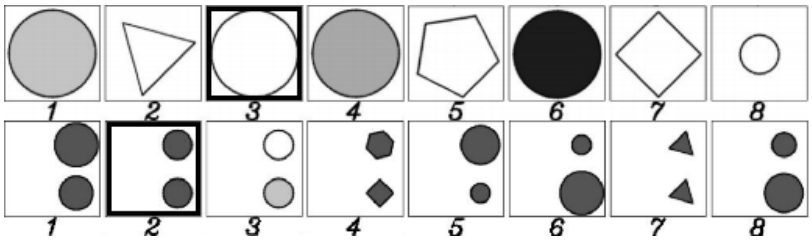


FIGURE 3.2: Two example answer sets from problems in RAVEN. We can derive the correct answer (emboldened) from each set by finding the intersection of the set's modes of shape, colour, and scale factors. Essentially, "which frame has the most common features?"

this, we realised that sufficiently powerful models could inadvertently exploit a statistical bias in the dataset, introduced by the sampling scheme used by the authors to generate the answer set of each problem. Note the following excerpt from the original paper:

> "To break the correct relationships, we find an attribute that is constrained by a rule... and vary it. By modifying only one attribute, we could greatly reduce the computation. Such modification also increases the difficulty of the problem." [174]

While this is an effective way of providing a challenging set with many plausible answers, it also provides a method of locating an answer context-blind. In other words, correct answers might simply be found by locating the mode over answer attributes, without even seeing the context frames. In Figure 3.2, we demonstrate that this is a simple enough strategy to be utilised by hand. To test this hypothesis, we trained models on the answer frames alone. In an unbiased set, the theoretical performance of such a model should be no greater than that of random selection in the long run; 12.5%, given a choice of 8 answer frames. On our solver, we were able to achieve an accuracy above 90%, averaged across all 7 problem configurations. Given that such performance over RAVEN is competitive with most current models, we confirm this as a significant issue potentially affecting a number of previous works.

This also impacts the reported generalisation ability of past methods; in our tests, locating the mode of a given answer set appears to be a skill that can be attained from one task and transferred to others, and we believe it to be an operation easy to acquire by the 1D convolutional module of our Rel-Base architecture (Section 3.6), given its task of finding local patterns between frame features from the first stage.

We wish to note to the community that we believe RAVEN to be a strong asset to our research, and we commend the original authors for their contribution. For its continued use as it is currently released, however, we believe that methods must process answer frames independently of each other, perhaps in a fashion similar to WReN. Therefore, the evaluation within some papers ([177], benchmarking

WReN in [173]) should still be correct, as their architectures already enforce this independent processing. Unfortunately, in [173], the model-level contrast summarizes common features within the answer set, and therefore misses this independence requirement. [179] also follows the methodology of [174]. This is of critical importance for the ongoing use of this dataset.

3.6 Architectures

In this section, we detail the three architectures benchmarked by our work. The purpose of our ResNet model is to serve as an analogue to the original in [174], in order to revise the literature with an accurate baseline. Our two novel architectures, Rel-Base and Rel-AIR, build on this simple network by adding additional encoding stages.

3.6.1 ResNet baseline

We use a 4-layer residual encoder with skip connections across pairs of layers, and stack frames into independent sequences – one per candidate answer – to be processed and scored. We borrow this design choice from [4], as it prohibits the model from comparing answers; this is in contrast to the original method, which processed all frames in a problem at once, one channel per frame. We set a kernel size of 7x7, stride 2, and spatial dropout (p=0.1) on all layers. We visualise this method in Figure 3.3.

3.6.2 Frame-relational ResNet (Rel-Base)

Improving on the baseline, Rel-Base encodes problems in two stages. The 4-layer encoder used in Section 3.6.1 first takes a batch of problems, embedding all frames individually. Embeddings are then stacked into candidate sequences as per the baseline method, and processed by a second encoder, consisting of 1D convolutional layers. In doing so, our model is able to learn a low-level perceptual process unaffected by the position of frames, and a higher-level that's tasked with modelling relationships by finding patterns in and between embeddings. Convolutional layers greatly reduce the number of weights compared to WReN's relation network [4], and we show them to be more data-efficient. Finally, Rel-Base does not require WReN's frame position vectors, as frame order is retained in the channel dimension.

3.6.3 Object-relational ResNet (Rel-AIR)

To solve this problem of generalising between problem configurations in RAVEN – i.e. to correctly process unseen object arrangements – it seems necessary to disentangle objects from their placement in a scene. Our full architecture, Rel-AIR, makes use of an initial unsupervised scene decomposition stage, AIR [33], which provides

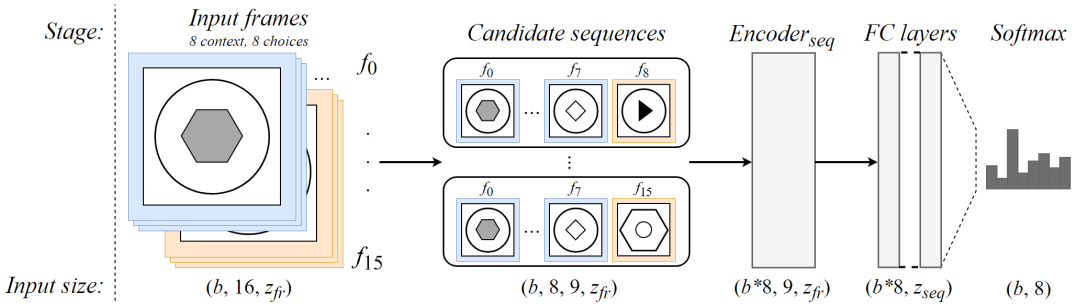


FIGURE 3.3: Diagram of the basic method. Given a batch of b problems, b*8 candidate sequences are formed, independently encoded, and scored. For Rel-Base and Rel-AIR, frame embeddings of size z_{fr} are generated by additional stages. For ResNet, raw frames are used.

an object-centric inductive bias. This is trained as a cascade architecture; AIR is first fit to the different configurations in RAVEN to extract objects, providing the training data for successive stages. Rel-AIR has five stages in total (see Figure 3.4 for a depiction of the first four):

1. **Scene decomposition**. The AIR module is tasked with observing all problem frames, and learning to decompose them into N object slots (with N being a predefined maximum, e.g. 9 slots for the `3x3Grid` configuration). Each 1-channel frame is therefore recorded as an N-channel image tensor, and an N-channel latent tensor detailing scales and x,y positions. In our experiments, we store both the contents of the attention windows and their reconstructions; while either can be loaded to train the following steps, we typically use attention windows. These slots are shuffled.

2. **Independent object embedding**. The 2D residual encoder then accepts a batch of objects and encodes them independently.

3. **Latent-informed object embedding**. The object embeddings from the previous stage are paired with their original scale and position latents, and a final conditional embedding is created by passing this paired data through a bilinear layer, in order to unify the two sources.

4. **Object-relational feature extraction**. The batch of object embeddings is reshaped into frames of N object channels, which is passed through a 1D residual encoder to generate the frame embeddings.

5. **Frame-relational feature extraction and scoring**. Finally, as with Rel-Base, these embeddings are stacked into sequences, encoded, and scored by fully-connected layers.

It is important to note that shuffling frames along the object dimension is critical to this model learning to make use of position and scale data, as we observed a strong correlation between the order of slots and their positions in the original image

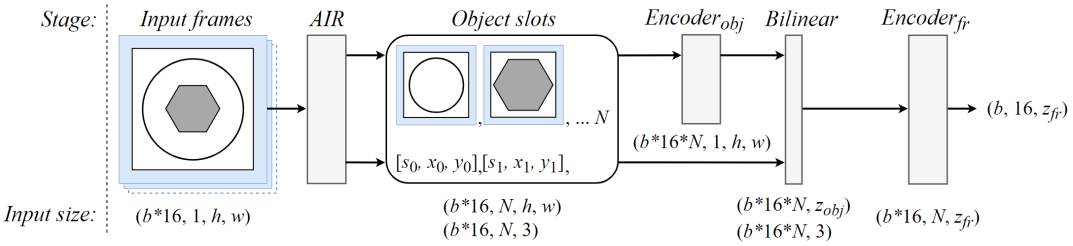


FIGURE 3.4: Frame encoding in Rel-AIR. The AIR stage decomposes frames into a maximum N constituent objects and their associated scales and x,y positions; s_n, x_n, y_n. Second and third, each object is embedded (size z_{obj}), and processed via a bilinear layer to incorporate latent data. Finally, each frame's object embeddings are convolved together, resulting in overall frame embeddings.

from AIR. Additionally, this shuffling operation promotes generalisation to problem configurations containing more objects than those trained on; without shuffling, only the first few frame channels would contain a signal, prohibiting the object-relational encoder from learning to use all channels.

3.7 Experiments

To evaluate the performance of our models, we make use of the aforementioned PGM and RAVEN datasets to test both overall (all tasks) and generalisation (cross-task) performance. To our knowledge, and given our findings in Section 3.5, only the WReN [173] and LEN [177] benchmarks for image-only RAVEN remain reliable in the literature. We train the three models described in the previous section, and use the same hyperparameters across both datasets. For reproducibility, we provide full details of these parameters in our supplementary material. Our code extends the official RAVEN public implementation[1], and is also available online.[2] Models are implemented in PyTorch [121] and Pyro [9].

3.7.1 Data

In addition to the commonly tested *neutral* set in PGM – containing 1.4 million samples with a 7:1 train-test split – we also use its challenging *extrapolation* set to more rigorously test model generalisation. To test performance over RAVEN-10k, we first train and test each model on the full set (consisting of all problem configurations; see Figure 3.5), before fitting models to individual configurations. We do not make use of the provided auxiliary information, we restrict image size to 80x80, or half-size, on both datasets, normalise pixel values to [0,1], and invert the dataset (to white shapes on black) so that the networks receive signal for shapes, not for the in-between space. Finally, we ensure training sets are shuffled, and make use of the same answer-set shuffling strategy as in [173].

[1] https://github.com/WellyZhang/RAVEN
[2] https://github.com/SvenShade/Rel-AIR

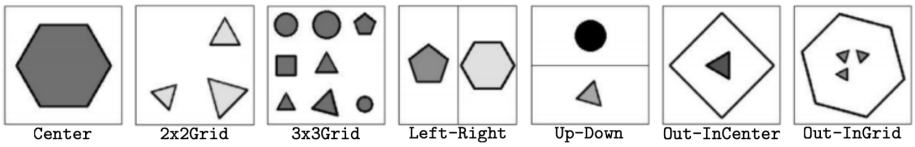


FIGURE 3.5: Example frames from RAVEN's diverse problem configurations.

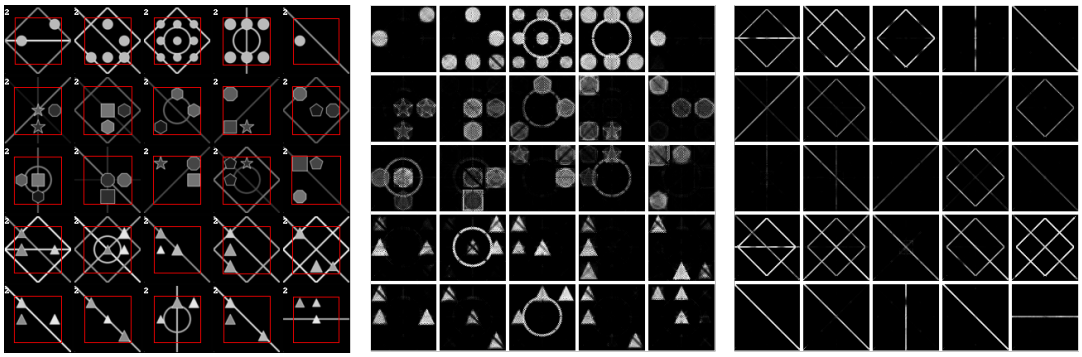


FIGURE 3.6: AIR decomposes PGM frames (left) into grid and background slots (centre, right). Red bounding boxes denote attention windows for the first slot.

3.7.2 Results on PGM

General performance.

We evaluate the overall accuracy of our first novel architecture, Rel-Base, using PGM *neutral*, and detail the results against existing image-only methods in Table 3.1. From this we notice exceptional performance; Rel-Base outperforms not only existing image-only models, but all models trained with the benefit of auxiliary data (excepting [57, 177], which achieve an extra 3ppt). This is an important result, as most other architectures are reasonably complex and specifically designed for RPM-style problem solving. Rel-Base instead offers a method that is agnostic to the problem setup, and can theoretically accommodate more general multiple-choice visual problems by changing the parameters of its stack function. Regarding data and training efficiency; we wish to also note that after a single epoch of training, Rel-Base reaches an average accuracy of 58.07%, exceeding what is reported by a fully-trained CoPINet.

While the Rel-AIR model is created specifically to improve performance across problem configurations, and therefore not benchmarked on PGM, we nonetheless preview the ability of AIR to decompose complex PGM scenes. In Figure 3.6, with two object slots, we notice that entities such as large background shapes and lines are separated from those that fall on the 3x3 grid, which is an encouraging preliminary result for future research.

PGM set	Wild-ResNet [4]	WReN	CoPINet [173]	LEN	LEN*	LEN**	Rel-Base
Neutral	48.00	62.60	56.37	68.10	70.30	85.10	**85.50**
Extrapolation	N/A	17.20	N/A	N/A	N/A	N/A	**22.05**

TABLE 3.1: Accuracy (%) of various models over neutral and extrapolation sets in PGM. LEN* and LEN** refer to the two-stream and two-stream with teacher model variants of LEN, respectively, as detailed in [177].

Method	Acc	Centre	2x2	3x3	L-R	U-D	O-IC	O-IG
WReN [173]	17.9	15.4	29.8	32.9	11.1	11.0	11.1	14.5
ResNet	34.5	41.7	34.1	38.5	33.4	31.7	34.6	27.3
LEN [177]	72.9	80.2	57.5	62.1	73.5	81.2	84.4	71.5
LEN+T [177]	78.3	82.3	58.5	64.3	87.0	85.5	88.9	81.9
Human [174]	84.4	95.5	81.8	79.6	86.4	81.8	86.4	81.8
Rel-Base	91.7	97.6	85.9	86.9	93.5	96.5	97.6	83.8
Rel-AIR	**94.1**	**99.0**	**92.4**	**87.1**	**98.7**	**97.9**	**98.0**	**85.3**

TABLE 3.2: Performance results of various models on the RAVEN set. We report accuracy (%) averaged across all configurations. L-R, U-D, O-IC and O-IG denote `Left-Right`, `Up-Down`, `Out-InCentre`, and `Out-InGrid` configurations, respectively.

Extrapolation performance.

We also test Rel-Base over PGM *extrapolation*, since to our knowledge, the literature has no other image-only model benchmarks for this task. We also want to verify that Rel-Base can exceed WReN here too, if we are to suggest that convolutional layers can be more widely adept at relational reasoning than WReN's explicitly relational architecture, e.g. pairwise operations over embeddings. We report these results in Table 3.1. From this, while we confirm the ability of Rel-Base to better generalise to the unseen factors in this set, we believe that properly handling this sort of extrapolation is a substantial research task that will require its own specific inductive bias, which is outside of the scope of this chapter. Yet, between both PGM sets, this strongly suggests that no utility is lost in the simpler architecture of Rel-Base.

3.7.3 Results on RAVEN

General performance.

We evaluate the overall accuracy of each of the three architectures, ResNet, Rel-Base and Rel-AIR, trained on the full RAVEN-10k set, alongside other image-only models, WReN [173], LEN and LEN+T [177]. We detail the results in Table 3.2, in which we demonstrate Rel-Base to be the first model to consistently exceed human-level performance on this task. Our full architecture, Rel-AIR, makes further improvements, beating the previous state-of-the-art [177] by 15.8ppt.

% of training set	ResNet	Rel-Base	Rel-AIR
10	14.79	24.40	**51.39**
25	21.48	52.24	**81.07**
100	34.51	91.66	**94.10**

TABLE 3.3: Accuracy (%) of models over RAVEN, given various training set sizes. Accuracy is averaged over all problem configurations.

Performance vs. training set size.

As in [173], we also explore model performance as a function of training set size, in order to further evaluate the efficiency of our methods. Table 3.3 reveals that, even with only 10% of the training data, Rel-AIR outperforms a fully-trained ResNet baseline. We believe Rel-AIR's strong performance is attributable to the AIR module's disambiguation of scene structure, alleviating the diversity of problem configurations by first resolving them to object lists.

Generalisation across configurations.

Finally, in order to properly test the ability of these networks to generalise, we replicate the format of Tables 4 and 5 in the RAVEN paper [174] and train all three methods on the following configuration regimes:

- Train on `Left-Right` and test on `Up-Down`, and vice-versa. As these configurations represent the transpose of the other, we expect models that have learned to understand notions of objects and object relationships to display reasonable transfer learning.
- Train on `2x2Grid` and test on `3x3Grid`, and vice-versa. Here, we're interested in the ability of models to apply knowledge across problems with fewer or more objects than they are familiar with.

It is important to note that we employed early stopping given validation performance *on the set to be generalised to*. Continued training adversely affected ResNet's performance, while Rel-AIR was least affected. Tables 3.4 and 3.5 detail our results. Firstly, we notice that Rel-Base and Rel-AIR both achieve accuracies significantly above baseline, indicating a strong ability to learn from limited data. Additionally, Rel-AIR displays a much higher proficiency in this task overall, often doubling the generalisation performance of Rel-Base. We also notice that ResNet performs much lower than random chance when generalising between `Left-Right` and `Up-Down`; interestingly, its average generalisation performance rises to just above random (13.65%), and dips when train and test configurations were the same (18.48%), when we didn't first invert the data. We imagine this is due to there being very little signal crossover between these configurations when images are white shapes on a black background; `Left-Right` and `Up-Down` objects scarcely overlap, and so the model overfits catastrophically.

	Left-Right			Up-Down		
	ResNet	Rel-Base	Rel-AIR	ResNet	Rel-Base	Rel-AIR
Left-Right	27.83	90.09	**98.07**	3.71	32.71	**66.77**
Up-Down	2.98	22.61	**60.81**	26.42	90.23	**94.84**

TABLE 3.4: Generalisation test between Left-Right and Up-Down configurations. Rows and columns indicate training and test sets respectively.

	2x2Grid			3x3Grid		
	ResNet	Rel-Base	Rel-AIR	ResNet	Rel-Base	Rel-AIR
2x2Grid	26.32	60.16	**88.24**	13.96	41.55	**67.01**
3x3Grid	14.36	34.03	**61.90**	33.84	68.16	**82.54**

TABLE 3.5: Generalisation test between 2x2Grid and 3x3Grid configurations. Rows and columns indicate training and test sets respectively.

As a simple ablation study, we also trained a position-blind Rel-AIR, replacing the bilinear layer with a linear layer. We notice that performance on both Left-Right and Up-Down configurations – and generalisation between them – falls to around $43\% \pm 3$; this is an intuitive result given the added ambiguity, since two populated object slots can refer to two different frames if the positions are unknown (e.g. a square on the left and triangle on the right, or vice-versa).

3.8 Discussion

Our first experimental outcome is the strong performance of Rel-Base in both datasets, which challenges the design philosophy of other work in this area, and hints at hidden ability in simpler, general purpose architectures. The second major outcome is Rel-AIR's ability to train and generalise even from a single task, which we accept as evidence in favour of its object-centric inductive bias.

There are some weaknesses that ought to be stated for the purposes of future work. As visualised in Figure 3.7, AIR sometimes clips large objects (usually triangles) – and while this didn't become an issue in testing, it still means the later stages of Rel-AIR receive sometimes inconsistent representations. This does become an issue with more advanced scenes, as we found out with Out-InGrid; AIR struggles to correctly decompose scenes with objects across significant size differences, and this isn't solved by simply increasing the scale prior's standard deviation. Instead, the centre grid is always encoded as a single 'grid object', which is an understandable abstraction, given the module has no prior understanding of shapes, and optimises for scene sparsity. Encouragingly, a number of recent papers have reportedly made progress on the robustness of AIR [27, 162, 149]; we expect that such improvements will minimise the need to fine-tune AIR between configurations.

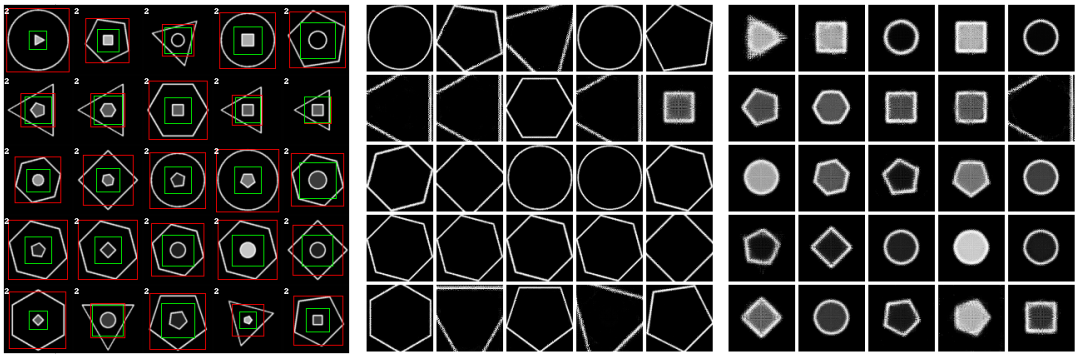


FIGURE 3.7: Visualisation of AIR's decomposition of `Out-InCentre` frames (left) into two slots (centre, right). Bounding boxes denote attention windows.

Another point worth mentioning is that, while the relational module never sees the type of task it is asked to generalise to, the AIR stage is pre-trained on each task. We believe this legitimises generalisation performance; as long as Rel-AIR remains blind to problems with novel arrangements of objects, it can be said to generalise its reasoning to them. As a future direction, the AIR stage might be trained by a scene generator that returns random arrangements of objects, which in turn, ought to aid with the 'grid object' failure case by providing increased diversity.

Finally, like other recent decomposition models [16, 54], Rel-AIR needs to be trained with the maximum number of object channels expected in a scene. This makes training over the full RAVEN set inefficient, as most tasks include far less than a full grid of 3x3 objects. Forming scene graphs (e.g. [171]) to be encoded via graph neural networks [138] represents a possible direction in handling the variable length outputs of AIR without padding them.

3.9 Conclusion

In this work, we have strived to enable neural vision models to perceive and compare abstract visual scenes in ways that permit generalisation between problem configurations. First, we navigated a critical issue arising from the answer-set sampling strategy in RAVEN, prompting our re-evaluation. We proceeded to show via a relatively general-purpose network, Rel-Base, that convolutional layers can learn to extract relational features more capably than existing architectures involving explicit relational operations. We have also shown that providing an object-centric inductive bias – via an unsupervised scene decomposition stage – makes further improvement over Rel-Base in generalising over RAVEN. Finally, models introduced in this chapter set state-of-the-art performance over both RAVEN and PGM datasets, despite the added challenges of using downscaled images and no auxiliary data, and invite a number of future directions at the intersection of scene decomposition and abstract reasoning.

	Acc	Centre	2x2	3x3	L-R	U-D	O-IC	O-IG
ResNet	83.11	84.23	65.34	68.70	95.14	95.82	92.02	80.53
Rel-Base	92.46	98.49	78.66	80.52	99.22	99.66	98.63	92.04

TABLE 3.6: Accuracy (%) of ResNet and Rel-Base, trained context-blind on RAVEN.

3.10 Supplementary

3.10.1 Context-blind performance

We list the results of context-blind variants of our solvers in Table 3.6. Note that these solvers have the stack function removed; sequence encoders are simply given all answer frames/embeddings instead. Rel-Base displays near-perfect performance in several configurations; this, as mentioned earlier, points to the importance of independent processing in comparing answer frames over RAVEN, if it is to be used to fairly assess the reasoning ability of these networks.

3.10.2 Model parameters

In Table 3.7, we provide all hyperparameters essential to reproducing the models introduced earlier in the chapter. This is intended to complement our code, released online[3]. All models were optimised with Adam, using *de facto* default parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a learning rate of $3e{-}4$. We used split-batch training to enable parallelisation across two GPUs, with a batch size of 32.

We wish to refer readers to the online repository of the Pyro library[4] as the hyperparameters set by this code were largely unchanged in our AIR module; we used a learning rate of 0.1 for the data-dependent baselines, a z presence prior of 0.01, and a decoder output bias of -2. Where we differed was in the scale prior's mean and standard deviation, μ and σ, which had to be fine-tuned for specific problem configurations in RAVEN. For the most part, we found $\mu = 2.0$ and $\sigma = 0.4$ worked well. For the `3x3Grid` set, pushing μ to 3.0 and σ to 0.2 was necessary to predispose the module to using small, regularly-sized attention windows – and therefore fill all 9 slots, instead of perceiving larger compound objects. As mentioned, AIR didn't correctly decompose `Out-InGrid`; instead, it produced 2 slots corresponding to the outer and grouped inner shapes.

[3] https://github.com/SvenShade/Rel-AIR
[4] https://github.com/pyro-ppl/pyro/tree/dev/examples/air

Module	Component	Parameters	
Convolutional block	Convolutional layer (1D or 2D)	Kernel size:	7
		Padding:	3
		Stride:	1 if 1D layer 2 if 2D layer
	ELU nonlinearity		
	Batch normalisation		
	Spatial dropout	Probability:	0.1
Residual block	Convolutional block		
	Convolutional block		
	Skip connection	+ max. pool if strided blocks	
Frame encoder (Rel-Base) & Object encoder (Rel-AIR)	2D Residual block	In/hidden/out:	1, 64, 64
	2D Residual block	In/hidden/out:	64, 64, 16
Frame encoder (Rel-AIR)	1D Residual block	In/hidden/out:	*N*, 128, 128
	1D Residual block	In/hidden/out:	128, 128, 1
Frame conditioning (Rel-AIR)	Bilinear layer	In-1/In-2/out:	400, 3, 403
	ELU nonlinearity		
Sequence encoder (Rel-Base & Rel-AIR)	1D Residual block	In/hidden/out:	9, 64, 128
	Max. pool	Downsample:	4x
	1D Residual block	In/hidden/out:	128, 128, 64
	Adaptive avg. pool	Size:	16
Sequence encoder (ResNet)	2D Residual block	In/hidden/out:	9, 64, 64
	2D Residual block	In/hidden/out:	64, 64, 64
MLP (All models)	Linear layer	In:	1600 (ResNet) 1024 (Rel-Base & Rel-AIR)
		Out:	512
	ELU nonlinearity		
	Batch normalisation		
	Dropout	Probability:	0.5
	Linear layer	In/out:	512, 1

TABLE 3.7: Details of the modules built for ResNet, Rel-Base, and Rel-AIR, along with their components listed in architectural order. Note that 'in/hidden/out' refers to the number of channels, and *N* refers to the number of object slots.

Chapter 4

Sharpening the Methodology Part I: Backing Models Into Corners

4.1 Introduction

Chapter 3 progressed three fronts. It provided two SOTA models, one of which can be thought of as an all-purpose PMP solver, providing a strong baseline for much of this thesis to come. It also uncovered a major exploit in RAVEN, one that was likely hidden due to the human tendency to assume that agents that come to similar conclusions have done so by similarly justified means.[1] Finally, it investigated inductive biases, including *objectness*, with which to pursue generalisation performance as an overlooked yet crucial test that RAVEN offers.

This chapter should be thought of as *Part I of II*, setting much of the theoretical and experimental groundwork towards the derivation of our own RAVEN variant in Chapter 5. Here, we continue to deepen the theme of dataset exploitation by assuming a relevant null hypothesis for this field of research: that models have *not* found the concepts that will allow for generalisation to OOD, due to the existence of unintended regularities in the data. We assume the responsibility of *backing models into corners* and leaving no way out, other than through our intended task. As the Impartial-RAVEN dataset was released subsequent to the last chapter [73], we decide to thoroughly test it against RAVEN in order to be confident in its abilities and limitations, before rendering it foundational to a new dataset. To serve these experiments, we appropriate the architectural design of Rel-Base to a set of baselines targeting potential exploits at multiple levels of abstraction.

Following on from this, we then discuss the nature of the RPM task when administered to humans, what kinds of reasoning this may involve, and therefore, what may be evaluated by human testing. We theorise the ways in which the RAVEN dataset alters this presentation for machines, changing the task and therefore, its evaluative utility. Through the lens of inductive inference, this discussion culminates with the proposition of higher-level exploits and a path to their resolution, thereby contributing to the nascent field of machine psychometrics.

[1] One part anthropomorphisation, one part false consensus effect [134].

4.2 Methodological changes

We begin this section with a sardonic assumption; if a highly-parameterised model is performant on a task that involves abstract reasoning, there exists an exploit.[2] While this may seem defeatist, this attitude is not dissimilar to assuming a null hypothesis — that a model has not found concepts that will generalise OOD — is the correct interpretation until enough evidence has been presented to warrant revision. Our proposed methodology embodies a stress-testing approach to the development of PMP solvers, proceeding to find breaking points in performance that can be used to identify critical training conditions. An underlying intuition is that, in discovering these conditions, we become far more aware of what our models actually require to accomplish their feats of reasoning. Said in metaphor, such conditions are leaks in need of patching, as we imagine model optimisation to move as water towards low resistance.[3]

Exploits

Potentially exploitable regularities in any dataset — let alone those involving abstract reasoning — may include features that can be memorised or otherwise recognised as statistically over-represented. However, by their nature, PMPs employ several distinct levels of abstraction, with each level potentially inviting leaks of its own. We propose the following baseline tests to be run in addition to evaluating models of interest, beginning with an inspection of class balance, and progressing upwards through these levels:

1. **Rule-attribute balance.** Does the dataset represent concepts evenly enough to avoid rewarding naive solutions (for example, blindly selecting dominant classes)? While less straightforward than checking for balance in traditional classification datasets — where there exists only one label per training instance, and that label denotes a particular class — this is achievable by deciding how to sort PMPs by the rules they instantiate. We do so by considering rules, rule combinations, and rule-attribute pairs.

2. **Frame hold-out.** Do there exist problems in both training and test sets that make use of the same frames? Is there a noticeable performance loss when this is disallowed, pointing to frame memorisation?

3. **Independent frames.** If a solver is trained to observe and classify each answer candidate independently — without observing the context — and performs above random chance, there exists over-represented features within those frames.

[2] Assuming also that these models are specialist (i.e. not foundation models) and have had no other means — either by massive exposure to other data, or possessing universal inductive biases — to acquire these kinds of processes.

[3] There is some theoretical basis for this metaphor, as neural networks — especially those with ReLU activations, which includes Rel-Base — are likely biased towards low entropy functions [115]. Unfortunately, this does not often match what humans would consider parsimonious.

4. **Context blindness.** Training a model to return a solution based on observing the full answer set, still without observing the context, can reveal exploitable regularities in answer set generation. This source of regularity was introduced in Section 3.5.

5. **Final row completions.** Training a solver to disregard the completed context rows can indicate if there are any over-represented row attributes, potentially short-cutting analogy-making. If the completed context rows are not required, then the PMP is not asking $Row_1 :: Row_2 :: Row_3$?, rather, it is prompting for a statistically sensible Row_3.

RAVEN and I-RAVEN

Subsequent to our investigation in Chapter 3, a derivative of RAVEN has been released, titled Impartial-RAVEN (I-RAVEN [73]). Employing a technique called Attribute Bisection Tree (ABT), I-RAVEN presents an altered answer set sampling strategy to circumvent its predecessor's known exploit. By selecting exactly three attributes to modify, starting with the true answer frame at the root of a binary tree and branching with each attribute modification, the answer set is filled with exactly $2^3 = 8$ answers, evenly balanced across attributes. We subject both datasets to further testing, in order to set up later sections of the thesis with confidence.

4.3 Architectures and splits

We continue to make use of the Rel-Base architecture introduced in Section 3.6.2, appropriating it to a set of general baselines with which to run all 5 of the above tests. These are described in Table 4.1. The *independent answer* and *no context* baselines observe answers alone, scoring them either independently or together, respectively. This facilitates baseline tests 3 and 4 specifically. *Last row* and *Rel-Base* solvers operate identically, although the former is only permitted to peek at the incomplete row, facilitating test 5. Architectural and training hyperparameters are unchanged from their introduction in Chapter 3, the only difference being that we train all baselines for 20 epochs. In our experiments, this was enough to achieve convergence.

To serve our experiments, we generate RAVEN and I-RAVEN splits of 30,000 problems each, as compared to the 10,000 problems as originally released, to allow for more thorough training when limited to a single configuration. We also generate an I-RAVEN split with frame hold-out enabled, which we achieve by restricting the assembly of training and validation/test problems to disjoint sets of frames. We use a train-validation-test ratio of 6-2-2 for all splits, matching the original work.

Scope

As it is infeasible to perform a breakdown of solver performance across multiple problem types, for multiple models, datasets, and configurations, we select RAVEN's

Baselines and components	Details
Independent answer	Scores each answer independently
Frame encoder	Accepts frames [8:]
Linear block (2 layers)	In: 400, Hid: 512, Out: 1, Reshape to 8, Softmax
No context	Processes answers together
Frame encoder	Accepts frames [8:]
Sequence encoder	In: 8 channels
Linear block (2 layers)	In: 1024, Hid: 512, Out: 8, Softmax
Last row	Scores each last row completion independently
Frame encoder	Accepts frames [6:]
Stack	8 sequences of 3 frames
Sequence encoder	In: 3 channels
Linear block (2 layers)	In: 1024, Hid: 512, Out: 1, Reshape to 8, Softmax
Rel-Base	Scores each full completion independently
Frame encoder	Accepts all frames
Stack	8 sequences of 9 frames
Sequence encoder	In: 9 channels
Linear block (2 layers)	In: 1024, Hid: 512, Out: 1, Reshape to 8, Softmax

TABLE 4.1: Details for the baselines used in our experiments. All are derived from Rel-Base, and therefore share the same components and layer dimensions unless stated otherwise. For reference, further architectural details, including frame and sequence encoders, appear in Table 3.7.

`3x3Grid` configuration exclusively. This scope permits us to hone our study to the one format, limiting simultaneous rules, while still remaining capable of representing all rule-attribute productions in the dataset. Limiting simultaneous rules is important in examining rule balance; if there are many rules in the one problem, it becomes less clear by which rules we should characterise that problem, and harder to be confident about which rules have been short-cut (this will become even more important in Chapter 6). Limiting the format also facilitates the work of Chapter 5 to follow, where we extend the `3x3Grid` configuration to allow for a number of Bayesian experiments. Finally, the instantiated rules, rule-attribute pairs, and combinations ($n \leq 3$), are stored with each problem to promote the fine-grained analyses below.

4.4 Shortcut hunting

In this section, we report on our findings from three experiments: ***rule balance***, ***rule performance***, and ***answer set generalisation***, targeting all of the points above.

4.4.1 Rule balance

Table 4.2 summarises the overall balance of rules and rule-attribute pairs, as present in our `3x3Grid` RAVEN and I-RAVEN splits. *Occurrences* reflect the number of problems in a split found to contain the rule in the left column. We also list rule combinations, specifically, problems that contain at least three instances of the same rule

	Occurrences		
Rule	RAVEN	I-RAVEN	I-RAVEN-HO
Rule (cumulative)			
`Arithmetic`	17950	18014	17275
`Constant`	20535	20265	24196
`Distribute-Three`	21906	21992	20963
`Progression`	21856	21849	21439
Rule (3x instances)			
`Arithmetic`	548	525	424
`Constant`	1339	1359	1695
`Distribute-Three`	2066	2061	1409
`Progression`	2001	2092	1668
Rule-Attribute pair			
`Arithmetic-Color`	7523	7435	6938
`Arithmetic-Number`	4294	4280	3895
`Arithmetic-Position`	4303	4337	4455
`Arithmetic-Size`	7487	7568	6951
`Constant-Color`	7615	7560	9337
`Constant-Number/Position`	4344	4182	4634
`Constant-Size`	7563	7504	9183
`Constant-Type`	10024	9864	12290
`Distribute-Three-Color`	7392	7546	6795
`Distribute-Three-Number`	4333	4220	3842
`Distribute-Three-Position`	4310	4317	3937
`Distribute-Three-Size`	7566	7496	6895
`Distribute-Three-Type`	9976	10072	8907
`Progression-Color`	7470	7459	6930
`Progression-Number`	4137	4318	4610
`Progression-Position`	4279	4346	4627
`Progression-Size`	7384	7432	6971
`Progression-Type`	10000	10064	8803

TABLE 4.2: Occurrences of rules, rule combinations, and rule-attribute pairs, as found in our generated splits. Each contains 30,000 problems. I-RAVEN-HO refers to the hold-out split.

type, to further add to our understanding of the dataset. While all three splits seem reasonably balanced, we do notice that rules operating on `Number` and `Position` attributes appear in problems roughly half the time. This is due to only one being valid per problem; changing `Number` necessarily changes `Position`. Additionally, the `Arithmetic` rule overall is slightly under-represented, as it is disallowed on the `Type` attribute. Importantly, the procedure we used to generate the hold-out split (denoted by *HO*) seems to have also subtly altered the balance of concepts, leading to all `Constant` rules being boosted. While we don't believe these findings to be problematic, they nonetheless provides a preliminary understanding of what concepts the datasets are representing.

4.4.2 Rule performance

We move on to examining the performance of our baselines. Looking to Table 4.3, we see the context-blind result recreated from the last chapter, as well as further validation that I-RAVEN's ABT strategy is effective, as the average *no context* accuracy drops from 79.34% on RAVEN, to 13.69% on I-RAVEN. Indeed, the performance of both context-blind solvers on I-RAVEN seem to indicate that contextual information

Baseline	RAVEN		I-RAVEN		I-RAVEN-HO	
Indep. Ans.	45.21	Dist3	14.90	Const	13.13	Const
	45.21	Arith	14.73	Arith	12.97	Prog
	44.86	Prog	14.12	Prog	12.95	Dist3
	37.55	Const	13.99	Dist3	12.38	Arith
Average	**43.28**		**14.33**		**12.79**	
No context	81.67	Dist3	13.93	Arith	13.69	Arith
	80.68	Prog	13.58	Prog	13.61	Prog
	78.99	Arith	13.26	Dist3	13.53	Const
	76.27	Const	13.24	Const	13.05	Dist3
Average	**79.34**		**13.69**		**13.43**	
Last row	67.87	Prog	40.08	Const	47.47	Prog
	67.80	Arith	39.66	Prog	45.56	Const
	64.26	Dist3	37.37	Arith	45.16	Arith
	62.94	Const	31.81	Dist3	36.61	Dist3
Average	**66.18**		**38.21**		**43.70**	
Rel-Base	81.53	Prog	84.95	Prog	87.26	Prog
	80.22	Arith	84.43	Dist3	87.04	Const
	79.85	Dist3	84.27	Const	86.29	Dist3
	78.36	Const	82.23	Arith	86.24	Arith
Average	**80.00**		**83.84**		**86.00**	

TABLE 4.3: Breakdown of model performance (% accuracy) by rule types for all four baselines. Models are trained and tested on `3x3Grid` splits from RAVEN, I-RAVEN, and I-RAVEN with enforced frame hold-out.

is required in order to perform significantly above chance, given that a random classifier for eight answer frames is expected to achieve 12.5%. We also find no performance loss going from I-RAVEN to I-RAVEN with hold-out, indicating that solvers are not able to exploit memorisation of individual frames. Perhaps surprisingly, we see the opposite — a ∼2% percent lift in overall performance — which we account for by remembering that the balance in the latter set is slightly skewed to simpler `Constant` problems.

A more significant finding is indicated by the performance of the solver limited to *independent answers*. Unlike *no context*, this solver does not have access to the shortcut uncovered in the previous chapter — which we shall refer to as the *answer set shortcut* from now on — and so we do not yet have an explanation for the ∼30% discrepancy.

Fine-grained performance

Prompted to dig deeper by the finding of an additional shortcut, we break down the results by combinations of rules, as communicated by Table 4.4. Looking at *independent answers*, we now observe that RAVEN's original answer set renders `Distribute-Three` and `Arithmetic` rules are far more exploitable than `Constant` rules. Importantly, this was not found by looking at a breakdown of problems by the four rule types in Table

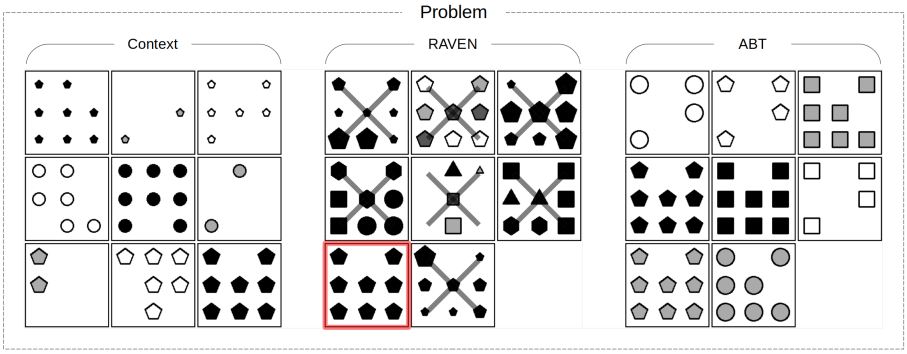


FIGURE 4.1: An extreme example of the intra-frame consistency shortcut as demonstrated by comparing answer sets on the same problem. By invalidating answers that display objects breaking attribute consistency, we are able to locate the correct answer with neither context *nor* cross-answer processing. Both shortcuts are patched by ABT.

4.3, as performance was averaged out enough to blur the discrepancy. So, what is happening here?

Both sampling strategies begin with the correct answer frame, and proceed to derive foil frames by altering one attribute at a time. While ABT ensures that answer attributes are balanced within problems, we believe it also results in a more balanced set of answer frames *across* problems. This is due to the fact that, for grid-configured problems (i.e. with multiple potential entities per frame, all facilitating the same set of rule-attribute pairs), most rules cannot be instantiated without *consistency* across an answer frame's entities. Unlike RAVEN, ABT does not modify the noise attribute `uniformity` (a decision made by the authors of [73] to keep answers more plausible when considering the context). RAVEN also presents more opportunities (7 vs 3) to disrupt intra-frame consistency, leading to there being less frames per answer set with this feature, raising the odds of selecting the true answer. These two changes account for the existence of this shortcut. Figure 4.1 demonstrates this further by displaying two different answer sets for the same problem. For clarity, while `uniformity` is unaltered in the particular frames shown in Figure 4.1, it generally contributes towards the shortcut by adding noise, perturbing consistent features, and "giving away" candidate frames as being more likely to be foils.

Table 4.4 also shows the effects of the unpatched shortcut impacting context-aware solvers. Theoretically, the rule requiring the most contextual information is `Distribute-Three`, as it becomes pure guesswork without the presence of a completed row. We see this result experimentally, as the *last row* solver performs the worst on I-RAVEN problems instantiating at least three `Distribute-Three` rules. This performance is tripled on RAVEN, with such problems continuing to be often solved by *Rel-Base* as well, strongly suggesting that even Rel-Base — an architecture designed to be unaffected by the known shortcut — was not fully immune to others.

Baseline		RAVEN		I-RAVEN	
Indep. Ans.	*High*	53.58	Dist3-Dist3-Prog	16.92	Const-Prog-Prog
		53.12	Arith-Arith-Arith	16.52	Arith-Const-Const
		52.02	Arith-Arith-Dist3	16.36	Arith-Arith-Const
	Low	32.10	Const-Const-Dist3	12.78	Arith-Arith-Arith
		30.78	Const-Const-Prog	12.68	Dist3-Dist3-Prog
		21.54	Const-Const-Const	12.62	Dist3-Dist3-Dist3
	Average	**43.28**		**14.33**	
No context	*High*	87.74	Dist3-Dist3-Dist3	16.23	Arith-Prog-Prog
		85.49	Arith-Arith-Arith	15.97	Arith-Arith-Arith
		85.29	Dist3-Dist3-Prog	15.69	Arith-Arith-Dist3
	Low	72.35	Const-Const-Prog	12.13	Arith-Arith-Const
		71.49	Const-Const-Const	12.04	Dist3-Dist3-Prog
		65.61	Arith-Const-Const	11.35	Dist3-Dist3-Dist3
	Average	**79.34**		**13.69**	
Last row	*High*	75.72	Prog-Prog-Prog	64.06	Const-Const-Const
		74.33	Arith-Arith-Arith	51.77	Const-Const-Prog
		72.72	Arith-Prog-Prog	51.66	Prog-Prog-Prog
	Low	62.02	Dist3-Dist3-Dist3	25.72	Const-Dist3-Dist3
		57.75	Const-Const-Dist3	25.57	Arith-Dist3-Dist3
		54.69	Const-Dist3-Dist3	20.66	Dist3-Dist3-Dist3
	Average	**66.18**		**38.21**	
Rel-Base	*High*	86.06	Prog-Prog-Prog	92.19	Const-Const-Const
		83.81	Dist3-Prog-Prog	87.61	Const-Prog-Prog
		82.44	Dist3-Dist3-Prog	87.52	Prog-Prog-Prog
	Low	76.20	Const-Const-Dist3	80.74	Arith-Arith-Prog
		75.50	Arith-Arith-Const	78.68	Arith-Arith-Const
		75.24	Const-Dist3-Dist3	73.54	Arith-Arith-Arith
	Average	**80.00**		**83.84**	

TABLE 4.4: Breakdown of model performance by rule combinations, for all baselines. High and low sections denote the top and bottom performing combinations. We do not perform this experiment on the hold-out split, as we have already confirmed that this does not add a source of exploitation.

4.4.3 Generalisation between answer set strategies

Curiously, the accuracy of our full model (Rel-Base) on RAVEN seems to be lower than on I-RAVEN, despite the aid of the frame shortcut being in place — and because this shortcut exists, we cannot be sure whether this discrepancy is simply accounted for by random variability between splits, without further testing. Given that we know: a) both of these splits are very closely matched in rule balance, and b) the frame shortcut is not patched by Rel-Base, we set up one final experiment for the chapter, to tie this loose end.

What answers challenge models?

Perhaps a more pragmatic question is, "how do answer set sampling strategies reward partially-informed guesses?" If a solver is unsure of the answer, yet is fairly confident that it knows the true value of one of the modified attributes, it ought to disregard all answers that do not possess that value. A suitable follow-up question is therefore, "what is the expected accuracy of a solver working with partial

information, for each sampling strategy?" If we imagine a frame embedding space consisting of one vector component per attribute, then RAVEN's original strategy presented solvers with a set consisting of foils tightly clustered around the correct answer (all with a Manhattan distance of one). If solvers were not, for some reason, able to learn the answer set shortcut, they would find themselves unable to wield their partial information to prune the answer set by much. For ABT on the other hand, we see that the average number of frames sharing attribute values, for any modified attribute, is 50%. Meaning that in the long run, a solver with confidence in the value of a single modified attribute would score 25%, pruning four answers and tie-breaking the remaining four by chance. This increases to 50% if it knows the value of two modified attributes, and 100% for three, as ABT only modifies three. So, from this perspective, we believe ABT presents a slightly easier set than RAVEN — albeit (likely) impossible to cheat.

Let us make a simple modification to the ABT algorithm — ATT, for Attribute Trisection Tree — which we depict in Figure 4.2. Here, we increase the branching factor by one, creating a ternary tree with the same depth. Since that change results in $3^3 = 27$ answers being generated, we keep the correct frame as before, and randomly sample down to eight. Like ABT, this allows each foil to have a higher Manhattan distance from the correct answer on average, lowering the odds of intersections of correct attributes occurring in the set, and making it easier to prune from an educated guess. Unlike ABT, which lowers these odds just enough to avoid being exploitable, we now see that the average number of frames sharing attribute values, for any modified attribute, is 33.4% before downsampling to eight answers. Simulating the average accuracy of a solver that has worked out a single rule gives us 36.8%, and with two rules, this rises to 75.2% in the long run.[4]

Testing brittleness

For the same reason that we designed Rel-Base to be unable to cross-process answers, ABT was invented as a means to patch RAVEN problems, at the cost of some expressivity. However, just because we do not want solvers cheating the task, does not mean we have to test them on a bulletproof set, *as long as they haven't learned its weaknesses*. A simple workaround for this is to train on problems with answers produced by either ABT or ATT, and test using the original, more challenging strategy. The first column of values in Table 4.5 shows the result of training and testing baselines on the ATT algorithm, experimentally confirming that this sampling strategy remains unexploited by blind solvers, but renders the RAVEN task far easier overall. Alongside these results, we record the performance of baselines tested on RAVEN after training on simpler answer sets. Observing a noticeable decline in performance as the generalisation gap is widened, we believe this strongly indicates that a) the

[4]To be clear, "the long run" was actually 10^6 simulated runs, which still takes less time for a computer scientist to code and run, than it does a mathematician to prove.

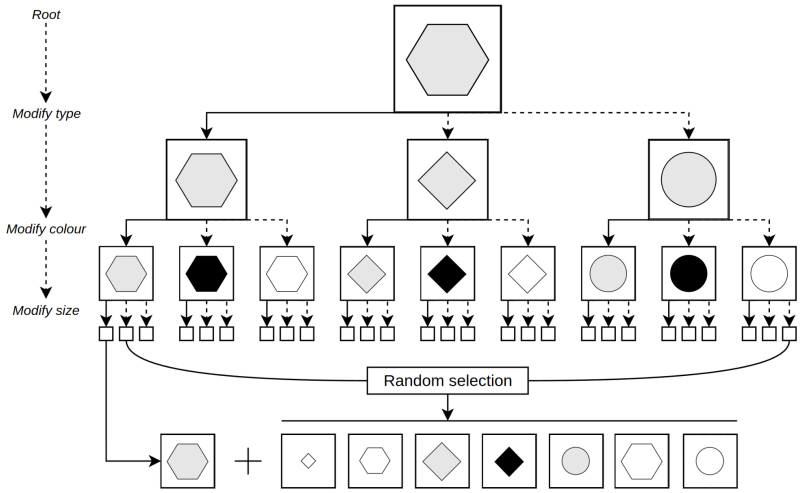


FIGURE 4.2: A simplified example of our ternary variant of the ABT algorithm [73], which introduces greater diversity to generated answer sets, but at the cost of decreasing problem difficulty. In this example, the root has been initialised with the correct answer, with the three randomly-selected attributes labelled on the left. As in ABT, child nodes are the result of cloning and modifying parents.

Baseline	ATT	ABT	ABT→R	ATT→R
Indep. Ans	14.88	14.33	4.82	5.49
No context	13.73	13.69	11.51	7.20
Last row	72.00	38.21	25.56	19.27
Rel-Base	96.05	83.84	72.60	43.50

TABLE 4.5: Generalisation between answer set strategies. Arrows denote shifts from training to testing. As predicted, while not exploitable by blind solvers, preparing a model on ATT alone leads to very high performance. This comes at the cost of overfitting, which is exacerbated as we decrease the spread of foils in testing.

true difficulty of the `3x3Grid` configuration was masked by the frame shortcut uncovered earlier, and b) there exists brittleness in solvers trained strictly with the one answer set strategy, which becomes more pronounced as the task is made easier. We expect these findings will be of use towards the development of solvers that can "do more with less", possessing the sample efficiency not to require exposure to the most discriminating answer sets in order to fully learn the task.

4.5 Higher-level exploits

In the first half of this chapter, we progressed through the levels of abstraction presented by RAVEN, systematically testing for breaking points. While lots of data were produced — and of course, not all will be equally useful — this process sets the tone of this chapter, encouraging this field to rigorously "do science" *on the tasks themselves* while remaining expectant of weakness. The findings so far indicate that I-RAVEN can generate suitable PMPs without significant cause for concern regarding exploitation. But there is an 'elephant in the room', a cognitive dissonance that arises from the knowledge that on this set, a) no state-of-the-art solvers up to this point are able to extrapolate without breaking, and b) state-of-the-art solvers otherwise appear to perform better than humans. We showed in Section 3.7 that the performance of all solvers, on both PGM and RAVEN datasets, severely declines when presented with a distributional shift. We saw this again in the last section, where generalisation between different answer distributions — even if the problem context and solution are identical — *still* results in noticeable loss. As remarked on in Section 2.5.2, the true goal of this area of research must be to build and evaluate systems that can demonstrate the dexterity required to navigate such shifts. There exists a hole in the literature.

In this half, we consider these critical, higher-level sources of regularity not yet commented on: what they look like, where they come from, how they may be situated in our problem space, and how we might bring them out of the woodwork. We therefore aim to recenter the spotlight of PMP research by more deeply understanding the task it presents.

4.5.1 The "problem" of induction

It is often taught that machine learning performs induction. While this isn't particularly controversial, it would benefit from specificity. The kind of induction expected of any of these solvers operates from an underlying frequentist assumption, which we will argue in this chapter may be enough in principle to approximate any function we want, but not able to reliably get there in practice.[5] *In the long run* is the phrase here, and indeed, the poor sample efficiency of deep learning approaches when compared with the routine inductive leaps of two-year-olds points to a key difference. As introduced in Section 2.4.2, the Bayesian approach to induction is considered to be more comprehensive when modelling human behaviour, as it appeals to the idea of prior belief as a complimentary source of knowledge to the observations. The Bayesian approach also shifts the game to one of distributing probabilities

[5] Although, there is nuance here too, as there are inductive biases present in both a randomly-initialised DNN, as well as the sampling made by stochastic gradient descent, which may be considered *almost* Bayesian [114]. This is an investigation for another thesis.

between hypotheses, which also captures uncertainty. Frequentism struggles to express themes of justification and certainty, but to some, this is insurmountable for any act of induction.

Criticism

As a philosophical concept, induction has endured criticism as being unable to provide incontrovertible justification, or to some, anything more substantive than conjecture. In the eighteenth century, Hume's fork was argued to separate synthetic statements from the realm of logic, claiming that science could not purport to hold *facts* about the natural world [29]. This is, to our mind, fairly uncontroversial too, as the phrase "*scientists have proven x*" is mostly a faux pas committed by the layperson. "*All models are wrong, but some are useful*" is a more apt aphorism [12]. Popper's denial of scientific induction (and attempted replacement with deductivism) is also in acceptance of the validity of such skepticism, aiming to restore the perception of science as a "rational" endeavour without it leaning on extrapolative inference [124]. Yet, the deductivist view does not capture the trajectory of science as it homes in on hypotheses likely to be worth testing, instead of exerting blind trial-and-error. For if we cannot point to some knowledge existing in the ways we approach hypothesis formation, how can the scientist rationally argue between untested hypotheses? Not just an issue for the scientist, but to Jackson's point from Section 2.1.1; how are we supposed to do *serious* metaphysics without a priori conceptual analysis? Here, we catch a glimpse at what more nuanced notions of inductive inference are trying to capture; *a navigational strategy that allows for a degree of efficiency and confidence towards the production of knowledge*. This is surely at work in the mind of anyone tasked with a matrix problem, but it can be said almost trivially from such a monolithic definition, so we shall proceed to flesh this out in the sections to come.

Definition

A more useful, working definition for induction might therefore be: *a process by which we confidently construct broad statements from specific statements*. These statements might be any concepts, allowing us to better perceive and predict the world from our wealth of past observations. Deep learning systems achieve this by using concepts — as presented by the system's current location on a very high-dimensional optimisation landscape — and navigating that landscape to tweak the parameters that underlie those concepts. But there is scarce *confidence* that the mode of landscape traversal is "rational", any more so than a Darwinian process [118]. This definition of induction also leaves out a notion of efficiency — and while deep learning can be considered many things — it has not once been called *efficient*. Clearly, there are many ways to perform induction; we have spent much of this thesis comparing different algorithms. At worst, one can fabricate all kinds of broad statements by

stream-of-consciousness beat poetry.[6] And while the outcome of this process can be said to have arisen from one's experience (trivially, since no thought enters *ex nihilo*, untethered from causality) this doesn't mean that one should expect such an outcome to be of much use. This sets up a chicken-or-egg scenario; how do humans efficiently obtain concepts with which to engage in inductive reasoning, from which new concepts will arise? The brain is doing something by only allowing certain ideas to pass into the spotlight of attention, filtering the torrent of data. Should we broaden our definition of induction to encompass this cycle, endangering us by reapproaching a vague and monolithic definition once more, or should we consider the obtaining of rules to begin with — the *fuel* of induction — as a process in its own right?

Abduction

To avoid spinning wheels, let us consider another form of inference: abduction. In its original sense, as proposed by Peirce [35], abduction drives the generation of theories to be subsequently assessed by other forms of inference. This also seems to at least gaze in the direction of analogical reasoning as we have discussed. Abduction may have been to Peirce, what analogy is to Hofstadter.[7] Peirce may have conceptualised this new form of inference by contemplating the way that concepts bring about others, causing truth to *flow uphill* as it were, from the specifics of the present, to the general. We might choose to picture this as perception resonating outward and sympathetically vibrating the vestiges of past scenes, as if via a shared fundamental frequency. From there, we can perform induction to refine and make sense of this resultant chord.

Moving then, into the modern sense of abduction, we directly incorporate the notion that some explanations will be far better than others (which is unrecognised by Peirce's work). This take on abduction is often characterised as "*inference to the best explanation*", and with this addition we begin to address both essential components of *efficiency and confidence* discussed a moment ago. To reuse the perhaps grandiloquent framing of analogy with resonant frequency, one could sit at a piano and strike any combination of notes — on some contrived grounds as to why they make sense together — but ultimately some will be tolerated by listeners far better than others. It is in this form of inference that we see many of the ideas from Chapter 2 re-emerge, recalling that the brain seems to be obsessed with both novelty and "elegance", as these are both hallmarks of the kinds of concepts that would do the brain well to digest, in its quest for predictive power. That the brain strives to maintain an orderly and fertile ground for producing useful concepts in the first place, instead of trawling a confused bowl of alphabet pasta, is the key intellective capability begging to be solved within the deep learning paradigm.

[6] The No Free Lunch theorem reassures us that there exists a world in which this algorithm is SOTA.
[7] This sentence is curiously, an analogy as to an analogy of analogy.

Putting it all together

While there have been countless attempts to distill the forms of inference and neatly depict their interplay, for the purposes of this thesis, we propose the nested optimisation process presented in Figure 4.3. Beginning at the base of the diagram, *deduction* tests the coexistence of present and hypothesised knowledge, and either accepts or rejects the new idea on the basis of logical entailment. In this sense, deduction is a formal and rigorous grounding force, allowing us to map out and navigate our knowledge in ways conducive to forward progress. Deduction will inform us of successful integration — or otherwise, of dissonance and incompatible truth statements — and will lead on to the other forms of inference to address its conclusions. Proceeding upwards, we engage abductive and inductive processes; this is the act of generalisation, running cognitive experiments that become less tightly coupled to observation and more creative the farther out they venture, ultimately finding novel connections. Finally, we return with new ideas in need of consolidation. In this depiction, there is a push-pull, from grounding in a formal world (math, logic), to moving away in two streams of conjecture: induction meditates on experience, weighing the probability of hypotheses in order to carry forward knowledge, while abduction ventures into creative ideation, seeking to resolve multiplicities and inelegance, pruning and refining the structures of thought, to then return with new hypotheses to ground once more. From inductive reasoning, we transfer properties from the observed to the unobserved, while abductive reasoning leads us to describe the *unobserved causes* themselves [141].

Overlaid on this diagram are regions that reiterate our understanding of the frequentist and Bayesian notions of induction. Bayesianism appears as a larger circle encompassing frequentism, but we have also let it encroach upon abduction, which is unconventional. We deliberately depict this boundary as blurred, as the next chapter builds a Bayesian approach to modelling RAVEN that appeals to tenets of explanationism; namely, that it incorporates inference to the best explanation in defining both prior and likelihood function.

Why have we called this a nested *optimisation* process? Observe abduction as the external loop: it must transmute observation into explanations that are likely to be of use to an agent. While humans are the inheritors of a rough set of neural weights, entrusted to us by millions of precedent lives, we also engage in meta-learning on an individual level. We learn how to learn, to find new ideas faster, within the particular memetic ecosystem we happen to find ourselves. In Figure 4.3, we suggest that abduction, as a function, optimises with respect to the behaviour of its inner loop, deriving its loss from the ability of induction and deduction to discern knowledge from conjecture. In this, it improves the way it filters raw observations and analogies. Induction and deduction take those explanatory hypotheses that have bubbled to the top and burst into conscious awareness, and — if they are still plausible after properly checking against present knowledge and observation — evaluate them, assigning a degree of confidence. Although not in these terms, the view embodied by

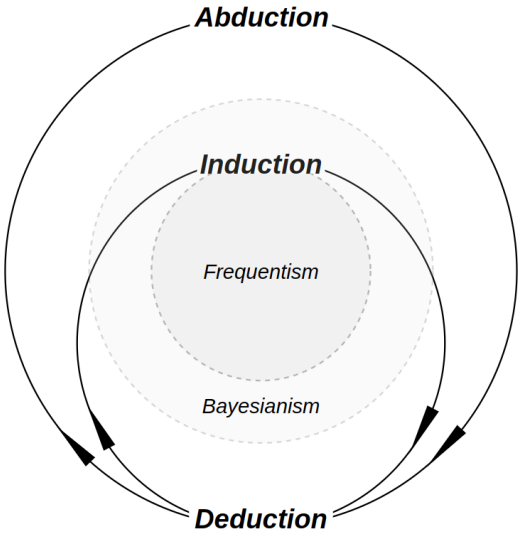


FIGURE 4.3: Diagram depicting inference as nested optimisation.

the "big data", or foundation model approach to AI, is that this optimisation is *not* nested — machines learn rules based on what works for the given data, guided by prediction. This approach appears severely limited. It does not learn how to uncover rules that might be useful to begin with, it relies on measurements between prediction and data as its only training signal, and it cannot understand or justify its rules, or wrestle them into structured cohesion — recall the discussion on the scattered nature of LLMs in Section 2.3.1). While there is nothing in principle to say that this approach will not achieve AGI with enough data, it should neither be thought of as a sophisticated nor safe avenue, as the closer it comes to mimicking intelligence, the worse the repercussions will be when underlying brittleness — only masked by data — causes catastrophic errors at the levels of responsibility entrusted to a seemingly competent and reasonable system.

Inference in PMPs

Kisielewska et al. claim that the process of hypothesis generation and testing involved in Raven's Advanced Progressive Matrices (RAPM) "essentially amounts to abductive reasoning" [89]. Let us therefore return to RAVEN, pointing to each part of this reasoning process as experienced by human solvers. Deduction is used to check the existence of hypotheses (e.g. `Progression-Size`). Given the finite and discrete nature of the problem space, each rule can be said to hold fully, or not hold at all. This is the only sub-task in RPMs that can be said to invoke logic, definitively responding "yes" or "no".[8] Armed with this diagram, we find the ability to

[8] Actually, one may test out such a hypothesis and respond with a "kinda". However, this should not be viewed as the outcome of deduction alone, rather, as responding "no" to the more general hypothesis, undertaking another inferential loop, and responding "yes" to a conditional variation of the original hypothesis. For example, while deduction might say that `Progression-Size` does not hold for all objects, the other forms of inference might dig deeper, presenting sub-hypotheses such as `Progression-Size-Large-if-Colour-White`, which may instead hold.

solve RAVEN problems after learning the rules of the game is not nearly the most interesting part — at that point, the inference machine works like clockwork, away from creative-abductive exploration and into routine, selective-inductive exploitation. For the machine model, having done the hard work of stumbling its way across an optimisation landscape to entangled features that happen to work, like Searle's Chinese Room [142] it can be imagined that such features, as over-parameterised and inelegant as they may be, are sufficient to lead towards a solution without any real understanding at all.[9]

Unfortunately, it doesn't end there. At this stage, if the PMPs themselves have an objective solution that arises from maximising the number of rules present — which we know is the case with RAVEN — then this reasoning process collapses to deduction alone. There is nothing more required of the model once it has found its parameters, no abductive or inductive reasoning, only rule classification. This is imperative to understanding why the extrapolation gap exists, and why our solvers display brittleness; something has changed in appropriating these tests for the machine. We have found our high-level exploit.

4.5.2 Understanding how our task has changed

How did we arrive here? Aren't *Raven's Progressive Matrices* supposed to test broadly for all kinds of inference?

For humans, tests in the RPM format become unreliable if participants are given coaching, or allowed repeated attempts and other kinds of practice [88], as this can "give the game away" when all problems instantiate rules from a relatively small hypothesis space [17]. But, we still believe that these tests can be of evaluative use in a machine learning context — and the argument is usually that such machines must learn all cognitive faculties *tabula rasa*, from the data alone — how else would we expect them to be performant?[10] So for humans, the very act of uncovering the right rules for a given problem *is* abduction, because they do not begin with a checklist of potential hypotheses to systematically validate. Instead, they engage in an iterative act of hypothesis construction and validation that we have tried to capture in Figure 4.3. For machine solvers, the format necessarily changes, and with it, the faculties these problem sets are able to test for. In RAVEN's `3x3Grid` configuration, a given problem may instantiate a total of 5 rule-attribute pairs, from a hypothesis space of 19. The task is therefore reducible to learning to encode each row in an embedding space capable of representing 5*19 features, and observing the similarity of embeddings within each completed context sequence. In other words, when

[9] That is, if we first understand *understanding* to mean more than "returning the right answer". Gettier cases demonstrate that justified true beliefs do not imply knowledge [50].

[10] Assuming we are not making use of foundation models, which is an idea that has only very recently seen application in this area (and with simplified text-based problems [164]).

this embedding is learnt, which is demonstrably achievable by these models after exposure to tens of thousands of problems, the task reduces to classification. The act of induction for the machine is in acquiring this embedding function, and not at test time, representing a divergence from the kinds of reasoning RPMs were built to evaluate. If we want to test for *generalisation from* prior knowledge, incorporating such knowledge via a roughly Bayesian process, induction should remain engaged in problem solving at test time. We therefore argue that *the RAVEN format needs revision for machine psychometrics*.

4.5.3 Induction and the problem space

Figure 4.4 introduces a space of problems and their solution strategies. $(A \cup B)'$ is the space of problems offered by RAVEN, which requires solvers to choose the answer associated with the largest set intersection of row-wise rules. That is, to maximise the number of rules shared by context rows. Problems in $A \cup B$ introduce ambiguity due to the number of valid rules not being enough to lead to a solution; solvers must weigh the presence of rules given other sources of knowledge. A solver successful at answering problems in crescent $A - B$ can do so by learning to rank rules, establishing some prior (in humans, this intuition of dominance might involve a notion of elegance as discussed, also considered "conceptual naturalness" by Tenenbaum [156]). Likewise, problems in crescent $B - A$ require weighing rules, this time by the context at hand. Problems in $A \cap B$ can be said to actively test for Bayesian induction, as isolated knowledge of rules, priors, or likelihoods do not present shortcuts to their solution. And while one cannot be sure that a solver hasn't optimised itself into an exotic conceptual space and found a new 'wrong' way to solve our problems, if we have followed a methodology such as the one built by this chapter, the odds are greatly decreased.

Note that our discussion delineating a kind of "full" vs. weaker forms of induction should not suggest that the former only occurs if both likelihood and prior sources of knowledge lead to a compromise (or even contradiction). Of course, making use of both sources is the Bayesian method, regardless of how informative either really is. However, it is the point of this section to introduce the notion of testing for richer forms of induction by *maintaining an attitude of shortcut-hunting*; while we could continue to predominantly test on a wide sample of problems, solvable by a number of strategies, *how would we know if the machine had used ours?* By first identifying a class of problem that permits no reliable solution strategy — other than the strategy we are wanting to evaluate — we get much closer to determining whether or not our models have learnt to reason like us at all.

We are aware that RAVEN is intended for use in deep learning, and therefore, it is unavoidable that the machine will become able to memorise the relatively limited set of rules and attributes. So for now, we leave abduction out of Figure 4.4, and focus on restoring induction to the task. Chapter 6 will bring this full circle, introducing a much more diverse problem space to begin tackling abduction.

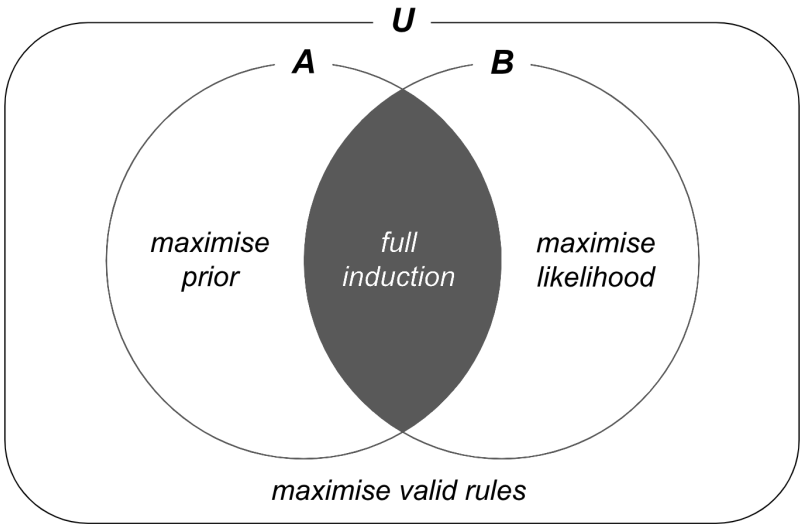


FIGURE 4.4: Euler diagram depicting a space of problems given valid solution strategies. While all problems require knowledge of rules, problems in sets A and B require prior and contextual knowledge, respectively, to rank such rules. Problems in $A \cap B$ can be said to actively test for Bayesian induction, as isolated knowledge of rules, priors, or likelihoods do not present shortcuts to their solution.

4.5.4 The utility of uncertainty

By proposing a Bayesian direction for RAVEN, we are willingly making a trade-off; we exchange RAVEN's objectivity and well-formedness for the promise of gaining insight into the thought processes of solvers. If we are looking for a solver's ability to achieve abstract inference in a roughly Bayesian manner, we can test for this by asking for reasonable inference as problems become ambiguous, presenting competing hypotheses that cannot be resolved by: a) maximising pattern stacking, b) picking patterns by the frequencies (or co-frequencies) for which they were seen in training, or c) otherwise learning a hierarchy of patterns, to say that some will win out over others regardless of problem context. We need to show that the model is required to learn something akin to hypotheses, priors, and likelihoods.

Although not presented by RAVEN,[11] rule ambiguity is still observed in some classical RPMs, not to mention the dominant role it plays in Bongard problems. Figure 4.5 illustrates such competition with a problem inspired by one of the later problems from *Raven's Advanced Progressive Matrices II* (we have made a derivative work to preserve copyright). The intended rule can be conceptualised as "lifting a curtain of lines" from two different angles *(answer 3)*. But there exists two other rules that both seem to justify foil answers:

[11] At least, not intentionally. The authors of [170] found that there does exist problems in RAVEN that are ambiguous by accident.

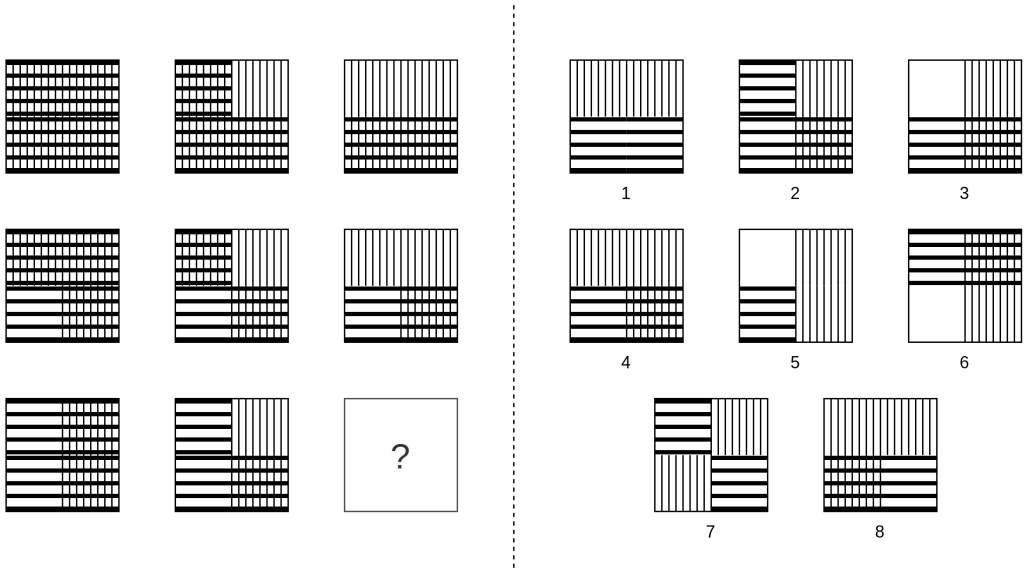


FIGURE 4.5: Reconstruction of an RAPM problem featuring rule ambiguity.

1. The first foil rule can be described as follows: "Consider three quadrant textures as distinct: vertical lines only, horizontal lines only, and cross-hatch. Perform progression over the number of instances of each pattern per frame" *(answer 1)*. Compared to the intended rule, this rule is less elegant.

2. The second foil rule: "Consider three quadrant textures as distinct. Start with the leftmost image in each row, and progressively overwrite the upper quadrants with the vertical lines texture" *(answer 4)*. This rule is less elegant again.

Incorrect answers can reveal a lot about a person's hypothesis space, and while we are not the first to suggest that analysing error patterns on RPMs can illuminate different approaches in reasoning [157, 106], generating strategically-ambiguous PMPs for machine psychometrics is a unique direction. The problem above is potentially more challenging and discriminative than it would have been, if not for the inclusion of compelling foils. To come full circle and revisit the *problem of induction*, one could say that there is no strictly logical reason why someone should select between the three of those possibilities. And yet, even with full knowledge of the rules, the addition of uncertainty forces us to make use of selective induction that reaches beyond this knowledge in search of resolution.

We argue that the original intended rule is still correct, as it leaves very little about the problem unanswered: one begins with a fully cross-hatched frame in the upper-left of the context, and by applying this hypothesis, precisely obtains all other frames. There is no surprise incurred, nor additional knowledge required. In contrast, the first foil rule leaves us without an explanation as to the organisation of quadrants in each frame. This might have been acceptable, if it weren't for the presence of a) some apparent organisational uniformity, which under this rule, becomes a *surprising coincidence* that clues us in to an undiscovered regularity, and b) a more

compelling answer frame. For the second foil rule, we cannot explain why the first frame of each row is initialised the way it is *at all*. Given this, we recognise the role of strategic ambiguity in the construction of problems, in order to test for inference that gets closer to the heart of what we truly want to measure.

4.6 Conclusion

We started this chapter galvanized by the findings of the chapter before it, acknowledging the reality of shortcut learning in neural networks. In the first half, we contributed a methodology targeting potential exploits at multiple levels of abstraction. This can be considered a type of feature importance analyses, used to reveal and eliminate confounding variables — *backing models into corners*.

Continuing to making broad use of our Rel-Base architecture, this investigation led to the discovery of another shortcut on the RAVEN dataset, as well as model brittleness when asked to generalise from one type of answer set to another, *even when given the exact same set of problem contexts and solutions*. With this result, we demonstrated that expanding the generalisation gap and exacerbating brittleness can be as straightforward as decreasing the average edit distance of foils to answers, between training and testing. We closed this section by contributing an analysis of the efficacy of different answer set sampling strategies, and called for the community to "do science" on the PMP tasks we assign to solvers, remaining expectant to find and patch sources of exploitable regularity.

The second half of this chapter took this further, squarely addressing the "elephant in the room" of this field of research — the extrapolation gap — and offered a deeper foray into inductive inference, its connections to abduction and deduction, and their interplay in PMPs. This culminated in a theory for the extrapolation gap — that *something has changed* in administering PMPs to machines, collapsing the problem to one of classification — as well as a path towards its resolution, seeking to introduce strategic ambiguity to the problem format and reinstating inductive reasoning at test time. We have therefore set the stage for Bayesian-RAVEN.

Chapter 5

Sharpening the Methodology Part II: Evaluating Inductive Reasoning with Bayesian-RAVEN

5.1 Introduction

With much of the groundwork set towards the derivation of our own RAVEN variant, this chapter forms *Part II* of an overhaul of the PMP methodology for deep learning models. We set out to construct *Bayesian-RAVEN*, a dataset to enable Bayesian experimentation with PMPs using the same rules and geometric primitives as RAVEN, but with competition between answers that cannot be settled by counting rules. This underlying uncertainty in answer sets continues the theme of *backing models into corners*, calling for the solver to integrate new sources of information — such as likelihood and prior knowledge — if it is to be successful.

We first devise a Bayesian model to serve as an oracle, returning a probability distribution over the answers to each problem it is shown. The results of this model, along with three ablations — likelihood only, prior only, and rule-stacking — comprise the labels of all PMPs in the dataset. We then perform a human study, enlisting participants to solve the same randomly-selected set of problems. Finally, we benchmark our machine solvers, maintaining use of the baselines defined in the previous chapter, to ensure we remain aware of potential shortcuts. By deliberately introducing and controlling ambiguity on the RAVEN set, and extensively comparing the performance of solvers, participants, and a novel Bayesian baseline, we take a step towards evaluating the inductive reasoning abilities of machines, understanding the conceptual hypotheses that they acquire, and their similarity to humans.

5.2 Related work

The *number game*

In his doctoral thesis, Tenenbaum championed a Bayesian model of concept induction that resolved the problems of the predominant rule-based and similarity-based

approaches of the time [156]. Both had failed to adequately capture how humans acquire hypotheses from only a few positive examples — sometimes, successfully from even one — generalising those hypotheses to new observations. If there was a *problem* of induction communicated in the last chapter, then this act of few-shot learning is on first glance, a *miracle*.

Key to the solution is recognising that Bayes' theorem presents an interplay of sources of information that can lead to decisions not explicit to the sources in isolation. On one hand, humans are guided by prior beliefs, displaying inductive biases towards what is *conceptually natural*.[1] On the other, we are always re-contextualising our knowledge, processing new observations in a way that minimises *surprising coincidence*. This is represented by the likelihood function.

While Tenenbaum applied his model to several tasks in his case studies, the *number game* is the one most applicable to ours. We summarise the game in the following preamble:

- *A set of computer programs exist, each entrusted with a simple arithmetical concept, e.g. "even number", "between five and ten", "power of two".*
- *Each program accepts an integer between zero and one-hundred, responding "yes" if this concept is present, and "no" otherwise.*
- *We have obtained for each program, a random subset of integers possessing that program's concept.*
- *From just observing each program's subset, the task is to then identify new integers expected to be met with a "yes" when given to that program.*

Tenenbaum compared the predictions of his Bayesian model against participants' responses, reproducing both rule and similarity-based generalisation, and therefore offering a unification of both. Participants were shown to induce clear mathematical rules when given compelling sets — {16, 8, 2, 64} strongly cues "power of two" — while they resorted to interval rules when shown less determinate sequences that didn't seem to follow anything else, such as {16, 21, 14} $\rightarrow$ "around 17".

Bayes and RPMs

At its core, the number game bears resemblance to the progressive matrix task. In RPMs, people are shown a set of rows (the *context*), knowing that each row presents one or more abstract rules. They are also given a set of frames with which to complete the last row, such that it may possess the same rules. In both tasks, humans are required to induce a concept from an exemplar set, and perceive that concept in a new instance. Despite this similarity, Bayesian modelling is uncommon for progressive matrices, and to our knowledge, the work by Little et al. [100] is the only

[1] The devil is in the details here: defining a useful prior is the most contested step in Bayesianism. As we have discussed, this controversy impacts machine learning as well, since there is no consensus on which inductive biases will lead to empowering a model, instead of restricting it.

investigation of this in the literature. They draw from previous work analysing the hypothesis space underlying the original RPMs [17], and in recognising that rules operate on individual features — such as shape type, colour, and size — they define a Bayesian model to compute posteriors over per-feature hypotheses, piecing together an answer for a given problem. Finally, they experiment with multiple prior distributions with which to fit their model to human performance data.

Since this work, the RAVEN dataset has been released, revitalising interest in modelling PMPs. Because this dataset was generated at the scale required for deep learning, such approaches have dominated this space [104]. However, we identify the Bayesian approach as being particularly complimentary; while the efforts of the deep learning community aren't invested in understanding humans to the same degree as the cognitive sciences, we are very aware that the tools we have to understand our machines are far from perfect. We wish to pursue two lines of modelling; one that builds machine learning architectures to solve complex problems, and the other that imports from cognitive science, encapsulating something about the human reasoning process that's likely absent from these architectures (and something we want to test for). The fact that RAVEN is a large synthetic dataset offers a number of advantages for Bayesian modelling. We have access to perfect annotations regarding the configuration of each problem, allowing us to experiment with more sophisticated likelihood functions derived in part from the statistics of the dataset itself. In return, once our Bayesian oracles are defined, we become able to label the problems in the dataset based on their outputs, establishing a new challenge.

5.3 Teasing out uncertainty in RAVEN

The problems in RAVEN were built to be *well-structured*, with each possessing a single, definitive answer. The original code does, on occasion, generate problems that accidentally relax this requirement. An estimated $\sim$5% of problems allow for competing answers [170].[2] While such ambiguity is usually thought of as undesirable, as we have established, there exists potential — if problems like these are used judiciously — to attain additional insight into a solver's reasoning processes.

To this end, we seek to alter the format of problems slightly, to increase rule uncertainty and encourage opportunities for prior knowledge to dominate contextual cues (and vice-versa). The goal here is to allow new answers to emerge that are the result of synthesising both sources of information. As an example of this interplay, we might expect to see the prior failing to extend to the particularity of a hypothesis, drawing away from overly complex rules. Simultaneously, the likelihood function might push for very unusual hypotheses, if the context could be considered surprising without their explanation. Humans hold both of these in tension, and so should these problems.

[2] We learned of this figure from correspondence with the authors, as it is not quantified in their published work.

Our main adjustment is to simplify the format to two context rows and four answer frames, making it similar to dedicated analogy problems such as letter string analogies (e.g. $a, b, c :: b, c, ?$ [70]) and the two-row visual analogy dataset introduced by [65]. To understand the effect of this change, consider the following sequence analogies, and their possible answers:

1, 2, 3 :: 2, 3, ? *(4 or 5?)*
1, 2, 3 :: 1, 3, ? *(4 or 5?)*
1, 2, 3 :: 3, 2, ? *(1 or 5?)*
3, 1, 2 :: 2, 1, ? *(3 or 1?)*

Without a second exemplar context row, we encourage "tie-break" scenarios to arise, forcing us to make use of our own intuitions about the strengths of competing hypotheses. Even on such simplified problems, we can start to see Bayes at work. In the first sequence above, `Progression` is a sensible hypothesis, as "1, 2, 3" seems to cue this fairly strongly. In the following sequence, "1, 3, ?" might cause us to reconsider `Progression`, as we would now need to imagine a shift from incrementing by one, to incrementing by two. `Arithmetic`, despite being slightly less intuitive, may be more appropriate in this context. Moving to the next sequence, `Progression` once again becomes a stronger hypothesis, for two reasons: `Arithmetic` would need a shift from addition to subtraction, which seems more contrived. Also, there is something *parsimonious* about the palindromic "1, 2, 3 :: 3, 2, 1", invoking ideas of conceptual elegance as mentioned earlier.

A problem context in Bayesian-RAVEN can be thought of as the visual representation of up to five such sequences combined. Figure 5.1 provides an example of this, mapping the four sequences above to colour, number, size, and type attributes, respectively.

We saw in the last chapter that the answer set sampling strategy played a critical role in determining a problem's difficulty and exploitability. So, the last element in constructing these problems is to generate answers that allow for rule competition, without re-introducing unintended regularity. Given our experiments with ABT and ATT, we are largely confident that finding eight answers as before (using ABT), before randomly downsampling (as in ATT) to four answers, will enable the construction of problems with varying degrees of ambiguity. Finally, we make several quality-of-life adjustments to the I-RAVEN code, including:

- Limiting the valid ranges of attribute values to simplify problems in the `3x3Grid` format. This was necessary as the only human experiment on RAVEN was performed with the original answer set sampling, which we are confident (although cannot experimentally confirm) also made the task significantly easier for humans, who were directed towards particular attribute values by looking to the answers. Switching to ABT sampling, trial participants often struggled to delineate between attribute values. We expect that the original authors

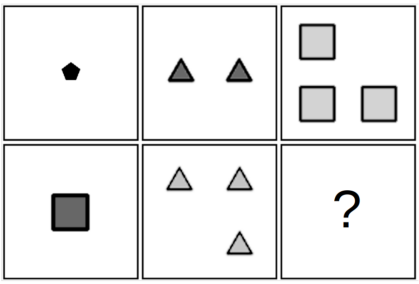


FIGURE 5.1: An example problem context, constructed from the ambiguous sequence analogies in Section 5.3.

chose six different possible sizes for entities to take, because this was fairly visible on most configurations. However, it is unsuited to a grid format where a given entity is restricted to a ninth of the frame. Likewise, distinguishing between two neighbouring shades when there are ten gradations of colour instantiated on very small shapes is almost imperceptible. Therefore, we only make use of four sizes, and four colours.

- Render sizes were also tweaked to be more perceptually matched between shape types, noticing that a circle may appear larger than a triangle for the same dimensions, given their different areas.

- We removed the noise attribute `Orientation`, setting it to zero degrees for all shapes. If left in, it would need to be accounted for by the Bayesian model. For this work, we set out to use the minimum set of rules and attributes with which to effectively carry out our experiments.[3]

- All problems were converted to a highly simplified string representation, anticipating future work with language models in this area.[4]

5.4 Creating a Bayesian baseline

We can now generate problems with no "objective" solutions. Therefore, in order to label them, we entrust four *oracles* to return reasonable targets. These oracles can be thought of as solvers that get to sidestep the perception stage, as they are given perfect knowledge of the hypotheses enabled by a problem and each of its answer frames (within the confines of the established hypothesis space). We define the following oracles:

- *Bayes.* This makes use of our full Bayesian model.

[3] *KISS* was a lesson learned by the author when his early and unweildy version of Bayesian-RAVEN pursued maximum functionality. This version had to be re-written from scratch.

[4] As an example, Figure 5.1 would be formatted as the following: "4203 3012-5012 0121-6121-8121 4122 0011-2011-8011". This is comprised of entity tuples (position, type, size, and colour ID), joined by hyphens to indicate frames, which are then separated by spaces.

- *Likelihood only.* In code, we implement this by calling the Bayes solver with a uniform prior.
- *Prior only.* This calls the Bayes solver with an uninformative likelihood.
- *Rule stack.* This uses both a uniform prior and likelihood, thereby scoring all hypotheses equally. This is capable of solving all RAVEN problems, choosing targets based on the number of rules they enable.

In this section, we motivate our design decisions for each component of our Bayesian oracle, including our choice of prior and likelihood, their integration, and the output of a probability distribution over answers.

5.4.1 Choosing hypotheses, priors, and likelihoods

Hypothesis space

Bayesian-RAVEN uses the same rules and attributes present in RAVEN. To turn this into a hypothesis space, we populate a three-layer hierarchy, starting with each of the four rule types: `Constant`, `Progression`, `Arithmetic`, and `Distribute-Three`. Building on these, the second level specifies the attribute that is controlled by that rule: `Number`, `Position`, `Type`, `Size`, and `Colour`. The final level describes the value of that rule, i.e. how it has been instantiated. Putting this together, the hypothesis `Progression-Colour-PlusOne` holds for a problem if the entities in a row become darker by one gradation per frame. If however, the entities in the first row darken by one, while those in the second *lighten* by one, we move to the more general hypothesis, `Progression-Colour`. As another example, `Constant-Size-Tiny` may hold in a problem that only contains tiny objects, but if the first row were to be swapped out with one containing only large objects, the hypothesis `Constant-Size` would hold instead.

We can remember that hypotheses take the form of *Rule-Attribute* and *Rule-Attribute-Value* tuples. In theory, we could generate even more general problems, where only a broad hypothesis like `Progression` might hold (an example of this would be a row incrementing by steps of two, followed by a row decrementing by one). In practice, we leave this option unexplored, as such problems risk becoming too arbitrary for humans or machines to confidently answer.

Prior

The work of Little et al. [100] defined two priors: the Carpenter prior, which was proportional to the "estimated ease with which each rule could be generated", and an accuracy-based prior, which made use of human data. Our primary goal is not to fully predict human behaviour *per se* — certainly, not in a way that is too tightly fit to our biases and perceptual idiosyncrasies[5] — although, we still ultimately want

[5]Consider: is it desirable for a machine to be fooled by optical illusions?

this model to be predictive of human responses. Instead, we want to measure inductive reasoning in machines, which requires building a prior from a more universal motivation.

As mentioned in Section 4.5.1, the RAVEN task requires deductive reasoning to check the existence of hypotheses. One possible way forward is to imagine the operation of a deductive program per hypothesis in the space. How easily does this program verify the existence of its hypothesis in a single row? What is the average number of steps taken for it to finish? How much memory is it expected to require? This use of complexity is not dissimilar to the Carpenter prior, which is itself an estimate of rule difficulty. Therefore, we decide to *roughly* estimate the algorithmic complexity of deductive programs as a nod in the direction of Solomonoff probability [75], as a way to push our Bayesian model towards incorporating *inference to the best explanation*, and as an early excursion into compelling territory.

We consider three basic functions with which to build programs:

Function	*Description*
`in(fi, mj)`	Checks if frame value f_i is at memory location m_j. Returns a boolean.
`put(fi, mj)`	Writes value f_i at location m_j. Returns *none*.
`diff(fi, fj)`	Calculates the difference of frame values f_i and f_j. Is non-commutative. Returns a new frame value.

These functions abstract away additional steps required to process the attribute values *within* a given frame. While we could write each program verbosely in machine language, for the sake of simplicity, we instead define k as a scaling factor to acknowledge the fact that some attributes are easier for the above functions to process. We assign k=1 by default. Hypotheses operating on the `Number` attribute are assigned k=2, recognising that enumeration of frame entities adds cost. `Progression-Position` and `Arithmetic-Position` are assigned k=3, as they involve even more costly entity-wise comparisons. Once we count the number of function calls made by each program, we multiply by this factor to obtain a complexity estimate.

We can now write a basic program to verify the existence of each rule:

Rule	*Program string*	*Function calls*
`Constant`	`put(f0,m0), (in(f1,m0) and in(f2,m0))`	3
`Distribute-Three`	`put(f0,m0), not(in(f1,m0)), put(f1,m1), not(in(f2,m0)), not(in(f2,m1))`	5
`Progression`	`put(f0,m0), not(in(f1,m0)), put(diff(f0,f1),m1), in(diff(f1,f2),m1)`	6
`Arithmetic`	`put(f0,m0), not(in(f1,m0) and in(f2,m0)), put(diff(f0,f1),m1), put(diff(f1,f0),m2), (in(f2,m1) or in(f2,m1))`	9

Each deductive program is passed the sequence of frame values relevant to its hypothesis attribute: for example, the program for `Progression-Colour` is given a

list containing the colours of objects in each problem frame. All programs begin by writing the initial frame, f_0, to memory. The last operation of each program involves a call to `in()`, and if it returns *true*, the hypothesis holds for that row. Our `Constant` program requires only three calls: having placed f_0 in memory m_0 and finding that both f_1 and f_2 are also already in memory m_0, it is able to verify its rule.

`Distribute-Three` needs five steps: it starts the same way as `Constant`, but places f_1 in memory as well, after realising that it is not already there. It then calls `in()` twice, to ensure f_2 is not in memory. Therefore, if `Distribute-Three` returns *true*, all frames in that row are differently valued.

`Progression` begins by confirming that the first two frames are different, before placing their difference in memory m_1. It then checks to see if the difference in the last two frames matches what is stored at m_1.

Finally, `Arithmetic` first checks to ensure the sequence isn't `Constant`, as without this check, the program would return *true* for sequences that fulfil $0 \pm 0 = 0$ (Bayesian-RAVEN does not label such sequences as `Arithmetic`). It then calls `diff()` twice, calculating both $f_0 - f_1$ and $f_1 - f_0$, i.e. the difference and sum of the first two frames, committing them to memory. Finally, it returns *true* if the last frame's value appears in memory.

These programs aim to provide a rough estimate of the steps required in deducing rules. However, given the numerous ways we may have conceptualised of this, does it matter that these estimates are defined somewhat arbitrarily? This very much depends on our purposes. Ultimately, this prior is for the Bayesian oracle to consider in labelling our dataset. We do not suggest that the encoding lengths of these programs succeed in approaching theoretical minima, nor do we define this prior as a benchmark for solvers outside of the microworld of Bayesian-RAVEN. We use this prior to aid an oracle in settling ambiguity, and we train our solvers to fit that oracle as a starting point, to see how they learn a *particular* way of navigating such ambiguity with the data they are given. Finally, we use this prior to help make predictions about human responses, because there is *parsimony* in the fact that it seems to line up with our intuitions of what humans find intuitive. In all, this prior — and the thoughts that went into building it — should be considered a placeholder, or skeleton, to be fleshed out by another thesis (and it might very well take an investigation of that magnitude). In Section 5.7, we will discuss opportunities to build in this very direction, including the use of algebraic machine reasoning [170] to construct priors, discovering concepts and estimating their complexities automatically.

Our full prior distribution is shown in Figure 5.1. To derive priors for hypotheses of the form *Rule-Attribute*, the scaling factor k for that hypothesis is multiplied by the inverse of the number of function calls in the associated program (less computational complexity leads to a higher prior). The distribution is then normalised such that it sums to one, representing each value as a fraction of the total probability mass.

Hypothesis	Calls	k	$p(h)$
Rule (cumulative)			
`Constant`	3	-	0.4473
`Distribute-Three`	5	-	0.2684
`Progression`	6	-	0.1905
`Arithmetic`	9	-	0.0939
Rule-Attribute pair			
`Constant-Position`	3	1	0.0994
`Constant-Type`	3	1	0.0994
`Constant-Size`	3	1	0.0994
`Constant-Colour`	3	1	0.0994
`Distribute-Three-Position`	5	1	0.0596
`Distribute-Three-Type`	5	1	0.0596
`Distribute-Three-Size`	5	1	0.0596
`Distribute-Three-Colour`	5	1	0.0596
`Constant-Number`	3	2	0.0497
`Progression-Type`	6	1	0.0497
`Progression-Size`	6	1	0.0497
`Progression-Colour`	6	1	0.0497
`Arithmetic-Size`	9	1	0.0331
`Arithmetic-Colour`	9	1	0.0331
`Distribute-Three-Number`	5	2	0.0298
`Progression-Number`	6	2	0.0248
`Arithmetic-Number`	9	2	0.0166
`Progression-Position`	6	3	0.0166
`Arithmetic-Position`	9	3	0.0110

TABLE 5.1: The prior distribution, showing the number of function calls associated with each rule, as well as a scaling factor, k.

Likelihood

To formalise our likelihood function, we first need to consider which generative model might be appropriate for explaining context sequences. In the work of Little et al. [100], a generative model is not considered, beyond whether an observed context is explicable by a hypothesis to begin with. In Tenenbaum's work [156], a generative model based on a *strong-sampling* assumption is chosen, i.e. that observations are drawn directly from the set belonging to the hypothesis. If we take X to be the observed problem context, defined as the set of attribute sequences, $X = \{x_{num}, x_{pos}, x_{typ}, x_{sze}, x_{clr}\}$, then the likelihood of observing attribute sequence x conditional on h is inversely proportional to the size of the set of all sequences contained in h. Tenenbaum called this the *size principle*: *smaller hypotheses that are nonetheless consistent with observations are more likely to indicate the rule for those observations*. At this point in the thesis, the connections between ideas like this and Occam's Razor, elegance, compression, and complexity should be clear.

Because RAVEN is a large, synthetic dataset, we are able to take a data-driven approach that would not have been feasible for the original RPMs, measuring the size of each hypothesis, $|h|$, from the problem set itself. This also allows us to define a more granular likelihood than our prior, finding values for specific hypotheses of the form *Rule-Attribute-Value*. However, since RAVEN problems involve stacking

several sequences together, they need to be separated first in order to measure $|h|$. As an example, consider a row where each frame contains one object. Regardless of the type or size of those objects, the corresponding sequence of `Number` values, x_{num}, is [1,1,1]. The hypothesis `Constant-Number-One` would hold for this sequence, and because we know there is only one sequence of `Number` values that instantiates this rule — i.e. $h = \{\{1,1,1\}\}$ — the size of h is 1. Alternatively, the less-specific hypothesis, `Constant-Number` also holds, which instead has $|h| = 9$, since $h = \{\{1,1,1\},\{2,2,2\},\ldots,\{9,9,9\}\}$.

To measure $|h|$, we find valid hypotheses for all attribute sequences in all problem rows in our dataset, and add those sequences to the relevant hypothesis sets (starting new sets as necessary). Therefore, $p(x|h) \propto 1/|h|$. To take this even further, we also incorporate a distinction between rule and similarity hypotheses, like Tenenbaum. To accomplish this, our likelihood considers further contextual information in the form of a similarity score, measuring how similar a candidate answer frame is to the rest of the row. By adding these functions, the Bayesian solver is aided in its ability to tie-break ambiguous problems:

- The similarity score, s, calculates the distance between a candidate frame and the mean value of the incomplete row. This is then normalised to [0,1]. For most attributes, this will be:

$$1 - \frac{\left|\frac{a_{f_1}+a_{f_2}}{2} - a_{f_3}\right|}{|a|}$$

  where a_{f_i} is the value of attribute a in frame i, and $|a|$ is the number of different values possible for that attribute. For the `Position` attribute, since frame values will be sets of positions, the similarity score is calculated as:

$$1 - \frac{\left|(a_{f_1} \cup a_{f_2})\Delta a_{f_3}\right|}{9}$$

  where Δ is the symmetric difference in position sets. This represents the number of shared object positions, out of a maximum of nine.

- As s is intended to help tie-break subtle cases, we raise it by a small exponent $\beta = 0.1$, to ensure it never dominates decision-making.

- The likelihood function therefore becomes $s^\beta/|h|$ when scoring hypotheses enabled by answer frames, and $1/|h|$ otherwise.

We display a table of reciprocal hypothesis sizes as recorded from the dataset, in 5.2. This table is truncated to include only simple hypotheses (*Rule-Attribute* pairs), as the full list contains 105,743 distinct hypotheses (most of which are variations on `Distribute-Three-Position`).

Hypothesis	$1/\|h\|$
Constant-Size	6.54E-05
Constant-Type	4.93E-05
Constant-Colour	4.78E-05
Constant-Position	3.55E-05
Arithmetic-Position	3.51E-05
Progression-Position	3.07E-05
Progression-Number	2.90E-05
Arithmetic-Number	2.47E-05
Progression-Colour	1.59E-05
Progression-Size	1.54E-05
Progression-Type	1.33E-05
Constant-Number	1.17E-05
Arithmetic-Colour	1.15E-05
Arithmetic-Size	1.10E-05
Distribute-Three-Number	9.84E-06
Distribute-Three-Colour	9.01E-06
Distribute-Three-Size	7.59E-06
Distribute-Three-Type	7.50E-06
Distribute-Three-Position	5.92E-06

TABLE 5.2: Reciprocal of hypothesis sizes, as measured from the dataset.

5.4.2 Applying Bayesian principles

In the last section, we defined each of our Bayesian ingredients, including:

- Observations (the problem context), defined as the set of attribute sequences, $X = \{x_{num}, x_{pos}, x_{typ}, x_{sze}, x_{clr}\}$
- Hypothesis space, $\mathcal{H}$
- Prior, $p(h)$
- Likelihood, $p(x|h) = s^{\beta}/|h|$ for hypotheses in answer candidates, and $1/|h|$ otherwise

The Bayesian oracle combines these ingredients in the following way:

1. The oracle is given a verbose list of hypotheses h that hold, both within the incomplete context, $\mathcal{H}_X$, and as enabled by each of the candidate answers y, $\mathcal{H}_{X,y_i}, i \in [1..4]$.
2. As an initial pass, the oracle removes the hypotheses that hold for all frames in the answer set, from both hypothesis lists, on the grounds that they are uninformative and work against the ability to discriminate between answers.
3. Following the work of Little et al. [100], the oracle proceeds to truncate $\mathcal{H}_X$ and $\mathcal{H}_{X,y_i}$ by keeping only a single hypothesis to explain each attribute. It does so by ranking posterior probabilities. Consider x_h as the attribute sequence relevant to h. The posterior is therefore given by $p(h|x_h) \propto p(h)p(x_h|h)$. We do not

calculate the marginal distribution of the data $p(x_h)$, as the utility of calculating posteriors at this stage is only for determining the strongest hypothesis for selection purposes.

4. At this step, if there are attributes left without a valid hypothesis to explain them, a placeholder is inserted, allowing for the similarity score to be still considered for that attribute in the absence of a better hypothesis. This is multiplied by a small epsilon ensuring it remains ranked lower than the rule-based hypotheses.

5. To calculate a probability for a given answer candidate y_i given context X, we follow the work of Tenenbaum: *"The probability of generalizing from the examples X to the new stimulus y is simply the ratio of the total score of hypotheses containing both y and X to the total score of all hypotheses containing X"*[156]. Therefore:

$$p(y_i|X) = \frac{\sum_{h \in \mathcal{H}_{X,y_i}} p(h)s^\beta/|h|}{\sum_{h \in \mathcal{H}_X} p(h)/|h|}$$

Because the above does not consider the probabilities of other answer candidates, the oracle returns a distribution over answers after normalising by the sum probability of all candidates.

5.4.3 Estimating confidence

So far, we have discussed the broad notion of *uncertainty* when discussing rule competition within a problem. Going forward, we wish to formalise *problem certainty*, δ_y, as the difference in probability between a problem's highest and second-highest candidates, as assigned by a solver. Because the outputs of solvers are normalised, δ_y falls in the range [0,1], with 0 indicating that the solver cannot break a tie, and 1 indicating that the solver is certain of a single answer. This serves as a simple measure of spread.

One thing that δ_y does not indicate, is the conceptual strength of a completed problem context. In other words, given two complete rows, can we estimate the strength of their relationship, and therefore use that to consider how compelling and well-formed the problem is? We need an understanding of strength to also recognise that, just because a solver found one answer to be far better than the others (expressing high certainty), *does not imply that it is a good answer*, or that the answer set may have almost entirely been unfit for the problem, or that the problem even *has a good answer in theory*. This is where the Bayesian framework is able to provide critical information, as it quantifies the strength of individual hypotheses. We define *problem strength*, ω_y, as the sum of hypothesis posteriors enabled by the highest scoring candidate: $\omega_y = max_{i \in [1..4]}(\sum_{h \in \mathcal{H}_{X,y_i}} p(h)s^\beta/|h|)$. We do this prior to both stages of normalisation (steps 5 and 6 in the last section), as we do not want to know how strong this answer candidate is in relation to the problem context (as the relations

in the context itself could be tentative), nor how competitive it is in relation to the other answers. We therefore consider problem strength to be the weighted sum of hypotheses enabled by the solved problem.

Building on both strength and certainty, we now establish a deeper concept, *confidence*, which reflects how thoroughly an oracle believes they have been able to solve a problem. If the oracle used is the Bayesian solver, which is responsible for the ground truth on this dataset, this can also be interpreted as an estimate of problem quality. We define confidence, c_y, as $\delta_y \omega_y$, capturing the following intuitions:

- A conceptually-strong problem offers a solution that enables strong hypotheses.
- A problem is solvable with high certainty if it presents a clear solution without competition.
- Confidence is the product of strength and certainty measures.
- *Confidence therefore estimates the degree to which an oracle has been able to unambiguously resolve compelling hypotheses.*

5.5 Method

5.5.1 General details

To benchmark Bayesian-RAVEN and to remain aware of potential exploits, we make use of Rel-Base and its blind derivatives, along with the same architectural and training hyperparameters as introduced in Chapter 4. To create the dataset itself, we first generated one large set consisting of 250,000 problems, along with partition files to instruct the program which problems to load. Each partition file defines a split — which is itself made up of problems from one or more problem types — containing 20,000 problems.

Problem types

Problems are labelled as belonging to four types: *basic*, *tie-break*, *new information*, and *induction*. These types fit neatly into the Euler diagram from the last chapter (Figure 4.4), each belonging to a different part of the space of valid solution strategies. The Bayesian oracle is considered foremost, and as such, all problem types require this solver to be able to provide an answer (only $\sim$5% of all generated problems are filtered out due to the Bayesian model not being able to disambiguate them). The following variables will be used to aid in defining these types: t_{solver} represents the target chosen by a given solver (its highest ranked answer candidate), while d_{solver} is the answer set certainty, thresholded as a boolean value. That is, $d_{solver} = \textit{true iff}$ $c_y \neq 0$, i.e. if the solver has been able to break a tie between answers. If $\neg d_{solver}$, then t_{solver} should not be considered a valid solution, even if it *happens* to align with

Problem type	*Logical definition*
Basic	$d_{\text{stack}} \wedge (t_{\text{stack}} = t_{\text{bayes}})$
Tie-break	$\neg d_{stack}$
New-info	$\neg d_{stack} \wedge ((t_{\text{bayes}} \neq t_{\text{prior}} \vee \neg d_{\text{prior}}) \vee (t_{\text{bayes}} \neq t_{\text{likel}} \vee \neg d_{\text{likel}}))$
Induction	$\neg d_{stack} \wedge ((t_{\text{bayes}} \neq t_{\text{prior}} \vee \neg d_{\text{prior}}) \wedge (t_{\text{bayes}} \neq t_{\text{likel}} \vee \neg d_{\text{likel}}))$

TABLE 5.3: Defining four problem types in Bayesian-RAVEN.

the Bayesian oracle. In other words, we consider the knowledge generated from an `argmax` when $\neg d_{solver}$ holds to be unjustified knowledge. With this said, problem types are defined in Table 5.3, which we describe as follows:

- *Basic* problems are solvable by rule-stacking. Formally, this is considered to be when the *Rule stack* oracle possesses justified knowledge of the solution found by *Bayes*. These can be considered traditional RAVEN problems.
- *Tie-break* problems are observed when counting valid rules leads to a tie, i.e. $\neg d_{stack}$.
- *New information* problems are a subset of *Tie-break*, with the additional requirement that the Bayesian oracle disagrees with either *Prior only* or *Likelihood only* oracles (or if it does not, then at least one of them must not be capable of breaking the tie in isolation).
- *Induction* problems are a subset again, identical in definition to *New information*, but enforcing that the Bayesian oracle is able to return a target that none of the other oracles have justifiably found.

Splits

With an understanding of our problem types, we create several different splits to serve a variety of experiments. Dataset splits differ by the ratios of problem types they include across training and testing, which we summarise in Table 5.4.

5.5.2 Experiments

Following is a list of experiments, along with their guiding research questions:

1. **General performance.** We measure the accuracy of solvers, blinds, and oracles, on general splits: *basic*, *neutral*, and *tie-break*.

   Research questions: How easily do our machine learning baselines fit to the Bayesian oracle behind our new dataset? Does their performance decline as we increase problem ambiguity? Are the problems in Bayesian-RAVEN largely solvable by humans? Are the problems in Bayesian-RAVEN exploitable by targeted baselines?

Split name	*Train → Test types*	Other notes
Basic	Basic → Basic	Presents a similar challenge to classic RAVEN.
Tie-break	Tie-break → Tie-break	Trains and tests on problems not solved by rule-stacking.
Neutral	All → All	Balanced sample of Basic and Tie-break problems.
Prioritisation	All → New-info	Tests prioritisation of prior or likelihood.
Induction	All → Induction	Tests integration of prior and likelihood
Induction-Plus	All (w/o New-info) → Induction	Tests both integration and generalisation.
Generalisation	Basic → Tie-break	Tests generalisation to problems not solved by rule-stacking.
Neutral-HC	All (HC) → All (HC)	Presents problems considered high *certainty*.
Brittleness	All (HC) → All (MC)	Tests generalisation to medium certainty problems.
Brittleness-Plus	All (HC) → All (LC)	Tests generalisation to low certainty problems.
Human	Basic (HC) → All	Used for the human experiment.

TABLE 5.4: The dataset splits partitioned for experimentation on Bayesian-RAVEN. As in Section 5.4.3, *certainty* is simply the difference in score between a problem's highest and second-highest candidates, as ranked by the Bayesian oracle. High, medium, and low certainty problems are found by sorting all problems and partitioning evenly into three.

2. **Predictive performance.** We compare oracle predictions to human data, including the number of problems correctly predicted, as well as the Jensen-Shannon distance between probability distributions.

   Research questions: How predictive are our oracles of human responses? Which oracles are more useful in solving different types of problems? What can we learn from directly comparing the probability distributions of oracles and humans, in addition to comparing overall accuracy?

3. **Prioritisation and Induction.** We measure the accuracy of solvers, blinds, and oracles on *prioritisation*, *induction*, and *induction-plus* splits, given different amounts of data.

   Research questions: How capable are baselines at learning the induction task when they are not *backed into a corner*, so to speak? How does the performance of baselines degrade as they are data-limited?

4. **Generalisation and signal utility.** We investigate the utility of different training signals on the *generalisation* split, given different amounts of data.

   Research questions: Do baselines acquire better concepts when trained on distributions of targets, instead of one-hot vectors?

5. **Brittleness.** We measure the performance of humans and baselines on the *neutral-HC*, *brittleness*, and *brittleness-plus* splits.

   Research questions: Bend or snap? How gracefully does the accuracy of humans and machines degrade as the uncertainty of problems increases between training and testing?

6. **Confidence.** The Bayesian approach has allowed us formalise a measure of confidence for a given problem. We calculate the inter-rater reliability between this measure and the self-reported confidence from participants.

Research questions: How does our formal measure correlate with confidence scores as rated by humans?

Human experiment

We employed 30 subjects using the Prolific.co research platform, who were remunerated at the hourly rate of Australian minimum wage. We requested a gender-balanced sample, with no other selection criteria. Subjects were presented with short instructions as to the structure of problems they would encounter, and familiarised with an initial set of 10 pre-solved problems, sampled randomly from the set of basic (high-certainty) problems. They were then instructed to complete a set of 32 test problems. To be a fair representation of our dataset, this was sampled randomly from the full set, with the only requirement being that each of the problem types needed to appear at least once. The same set of 32 test problems was shown to each participant, with the order of problems randomised per experiment to avoid confounds. For each problem, after clicking on a chosen answer candidate, humans were immediately asked to rate their confidence in their selection, before proceeding to the next problem. Confidence was measured on a 3-point Likert scale, with points labelled "low, medium, high". Our experiment was designed using PsychoPy [122] and hosted on Pavlovia.org.

As all participants were shown the exact same test set, we were able to obtain a probability distribution over the answers of each problem by summing human votes and performing normalisation, as per the oracles. Confidence scores were similarly found for each problem: we converted each Likert rating of [1..3] to a decimal between [0,1], and took the mean of human responses. As there were no problems that perfectly split the vote of human participants, the targets were chosen by majority.

Finally, no fine-tuning of the Bayesian model occurred aside from balancing the influence of prior and likelihood terms with exponents, to ensure that neither source of information dominated predictions. We used an exponent of 0.2 for the likelihood, and 0.5 for the prior.

5.6 Results and discussion

5.6.1 Visualising oracles

To begin understanding the decisions of our oracles, and to start qualitatively discussing their reasoning when compared to humans, we first visualise their outputs as simple heatmaps over problem frames. This is achieved by overlaying black frames on each answer candidate, performing feature scaling to obtain answer scores in the range [0,1], and setting the transparency of black overlays equal to their corresponding score. This allows us to gauge rough preferences and uncertainties for given problems at a glance. We also generate a red bounding box to highlight the

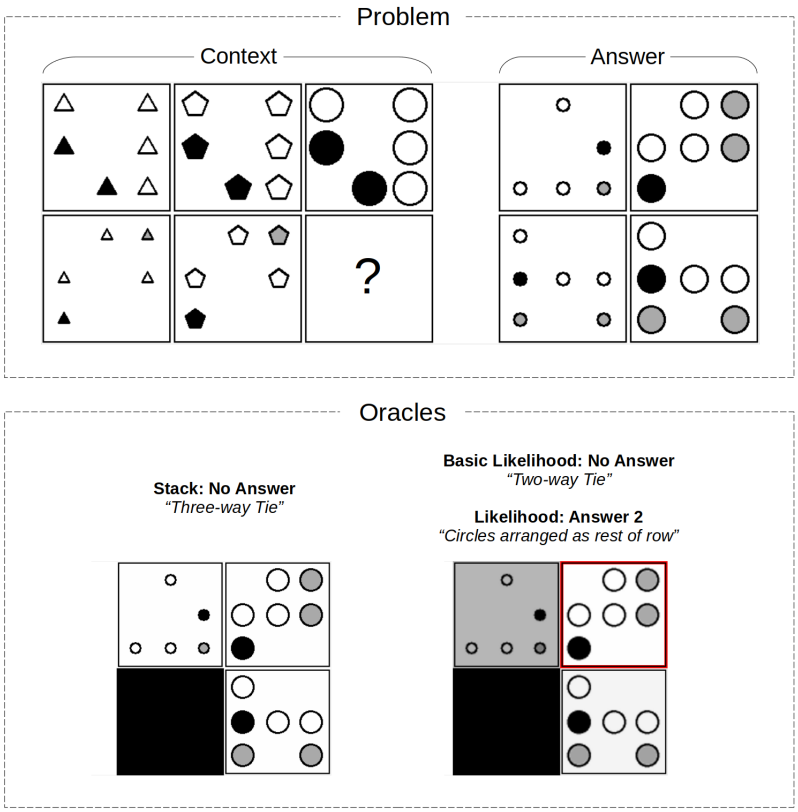


FIGURE 5.2: An example of an ambiguous problem in Bayesian-RAVEN requiring contextual cues to solve. Our likelihood function resolves a three-way tie by incorporating knowledge about the diversity of each instantiated hypothesis (i.e. the number of ways that particular instantiation might have looked), as well as additional context supplied by the first two frames of the incomplete row. Answers are represented as heatmaps, overlaid on the answer set.

target answer, in the case that the visualised solver has been able to disambiguate one.

Example 1: Visualising likelihood

The generated problem in Figure 5.2 presents as ambiguous to a rule stacking strategy, which results in a three-way tie. Here, a satisfactory answer is returned by the oracle considering likelihood alone. To locate an answer, it rates two of the three tied candidates equally higher, on account of the size of the hypotheses they enable. Finally, it resolves the two-way tie by 'noticing' that the arrangement of objects in Answer 2 more closely resembles the arrangements of objects in the incomplete row. In doing so, the Likelihood solver demonstrates the ability to disambiguate some problems by incorporating extra information in the context.

Example 2: Visualising Bayes

The generated problem in Figure 5.3 exemplifies the space of problems at the centre of our Euler diagram (Figure 4.4). This figure communicates: a) the ambiguity of this

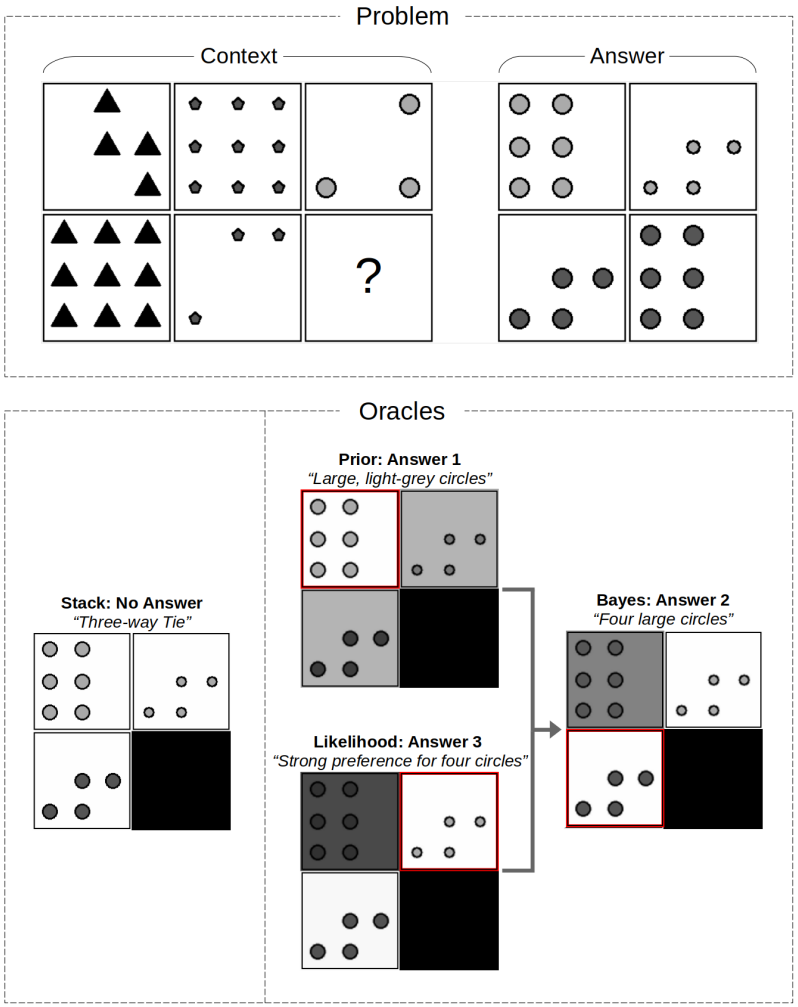


FIGURE 5.3: An example of a highly ambiguous problem in Bayesian-RAVEN, requiring integration of prior and contextual knowledge to break a three-way tie. The responses of all four oracles are represented as heatmaps.

problem, revealing the fact that, again, a rule stacking approach will lead a solver to a three-way tie, and b) the process of induction, by which our Bayesian model selects rules and scores frames by integrating likelihood and prior knowledge. The Bayesian model breaks the three-way tie and returns an answer that is not returned by the other three oracles. Table 5.5 provides further insight into the behaviour of the oracles on this problem, breaking down their selected per-answer hypotheses.

Example 3: Bayes and human participants

Figure 5.4 displays another induction problem, this time, from the set of 32 problems observed by all human participants. It is both the most ambiguous problem in that set, and the only problem where the Bayesian solver returned a different result to both prior and likelihood recommendations. Here, we see the knowledge possessed by the rule-stacking oracle is completely inadequate, as all answers enable the same number of rules. Prior knowledge preferences rules contained

Hypotheses	$p(h\|X)$	*Hypotheses*	$p(h\|X)$
Prior			
Ans 1		*Ans 2*	
~~dist3_tpe_(1, 3, 5)~~	~~0.2428~~	~~dist3_tpe_(1, 3, 5)~~	~~0.2428~~
dist3_sze_(1, 2, 3)	**0.2428**	dist3_clr_(1, 2, 3)	0.2428
dist3_clr_(1, 2, 3)	**0.2428**	dist3_num_(3, 4, 9)	0.1717
Ans 3			
~~dist3_tpe_(1, 3, 5)~~	~~0.2428~~		
dist3_sze_(1, 2, 3)	0.2428		
dist3_num_(3, 4, 9)	0.1717		
Likelihood			
Ans 1		***Ans 2***	
~~prog_tpe_2~~	~~0.1353~~	dist3_num_(3, 4, 9)	**0.3007**
prog_clr_-1	0.1236	~~prog_tpe_2~~	~~0.1353~~
arith_sze_sub	0.1169	prog_clr_-1	**0.1236**
int_num	0.0500	int_sze	**0.0493**
int_pos	0.0480	int_pos	**0.0461**
Ans 3			
dist3_num_(3, 4, 9)	0.3007		
~~prog_tpe_2~~	~~0.1353~~		
arith_sze_sub	0.1169		
int_clr	0.0497		
int_pos	0.0461		
Bayes			
Ans 1		*Ans 2*	
~~prog_tpe_2~~	~~0.0300~~	dist3_num_(3, 4, 9)	0.0516
dist3_sze_(1, 2, 3)	0.0276	~~prog_tpe_2~~	~~0.0300~~
dist3_clr_(1, 2, 3)	0.0275	dist3_clr_(1, 2, 3)	0.0275
int_num	0.0025	int_sze	0.0025
int_pos	0.0024	int_pos	0.0023
Ans 3			
dist3_num_(3, 4, 9)	**0.0516**		
~~prog_tpe_2~~	~~0.0300~~		
dist3_sze_(1, 2, 3)	**0.0276**		
int_clr	**0.0025**		
int_pos	**0.0023**		

TABLE 5.5: Breakdown of per-answer hypotheses and their posteriors, as selected by oracles for the induction problem in Figure 5.3. Strike-through hypotheses are removed from decision-making, as they are shared by all answers. Hypotheses with the `int` prefix include the likelihood's `Interval` rules (inserted when no other rule is found on that attribute), allowing contextual knowledge of frame similarity to influence the decision process.

in the left column of answers (`Progression-Colour` has a much higher prior than `Progression-Position`). Contrariwise, likelihood pushes decision-making towards the right column, as it would be a surprising coincidence for so many objects to position themselves perfectly at random. Yet, the likelihood offers one piece of information that allows the Bayesian solver to tie-break; answer frame 1 is found to be more closely matched in size to the rest of the incomplete row, and is therefore

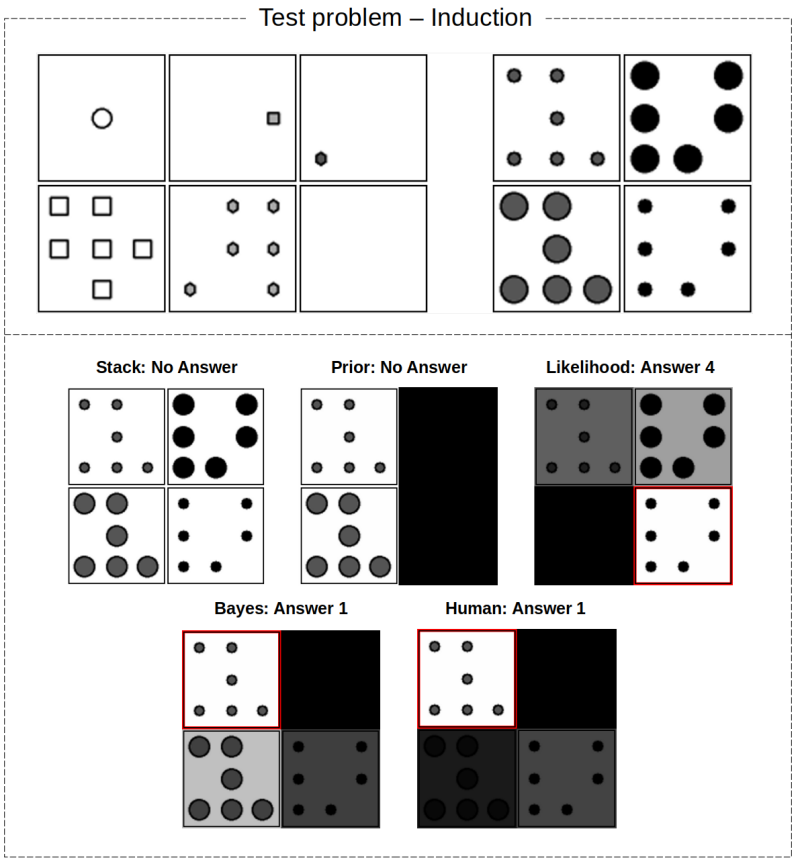


FIGURE 5.4: A problem in the human test set considered a four-way tie by rule-stacking. The output of the Bayesian solver closely matches human responses.

returned. Averaging all human responses on this problem generates a heatmap that closely aligns with the output of the Bayesian solver.

5.6.2 General performance of solvers

To begin our quantitative discussion, we first ask broad questions in order to establish our new dataset. We wish to know how easily our solvers fit the oracle behind Bayesian-RAVEN, while ensuring the problems themselves aren't exploitable by targeted baselines. We also verify that these problems remain largely solvable by humans.

In Table 5.6, we notice competitive performance from *Rel-Base* on the basic split, matching human performance. Moving to neutral, *Rel-Base* slightly under-performs the *Stack* oracle, which serves as a baseline to indicate the expected performance of a solver that has successfully learned to enumerate rules. As the difference between *Rel-Base* and *Stack* decreases when moving from basic to neutral, with *Rel-Base* ultimately outperforming *Stack* on the tie-break split, we hypothesise that it has acquired some features that are helpful in dealing with uncertainty. When compared to human participants, Rel-Base declines in performance much faster, as humans remain more adept at integrating information with which to resolve ambiguity. This

Split	Basic	Neutral	Tie-break
Solvers			
Rel-Base	84.58	68.29	53.73
Human	84.62	79.15	73.68
Blinds			
Indep. Ans.	28.40	27.12	27.10
No Context	28.58	27.34	26.92
Last Row	61.65	44.44	35.92
Oracles			
Stack	100.00	72.47	46.03
Prior	100.00	83.83	67.93
Likelihood	99.45	84.82	70.63

TABLE 5.6: General performance of solvers, blinds, and oracles, on Bayesian-RAVEN (% accuracy).

is an important result, as it is not revealed by testing on basic problems alone (i.e. problems in RAVEN and I-RAVEN).

Considering the performance of blinds, we notice that *Independent Answer* and *No Context* solvers stay within ±4% of random performance (25%) on all splits, which strongly indicates that Bayesian-RAVEN remains unaffected by these particular exploitation strategies. Meanwhile, *Last Row* is, on average, two thirds of the performance of *Rel-Base* in a given split, scoring as high as 61.65% on basic, and falling to 44.44% on neutral. This seems higher than on I-RAVEN, recalling Table 4.3 from the last chapter. Here, we recognise two effects. The first is the double-edged sword of removing a context row: while this increases rule uncertainty by encouraging competition between hypotheses, for basic problems that are defined by the absence of such competition, this change actually *decreases* the chance of requiring additional information. The second effect is from downsampling foils from 8 to 4: *Last Row* performance increased with the ATT sampling strategy in the last chapter in part for its downsampling from 27 to 8 foils. These effects explain the decline in *Last Row* performance as we decrease the number of basic problems, moving left to right across the splits. We do not find this to be problematic if it is recognised and accepted as a property of the dataset, especially given that the basic split is not intended to be focal.

Finally, the inclusion of oracles in this table serves to confirm that the dataset is set up correctly, with *Stack* solving all basic problems, and falling to just under 50% as problems present tied answers. For the latter problems, both prior and likelihood seem competitive with each other, indicating that they are both balanced in the operations of the Bayesian oracle.

5.6.3 Predictive performance of oracles

To better understand how predictive each oracle is of human responses, we measure both the number of problems the oracles predicted correctly, as well as the distance between answer distributions. The former serves to broadly indicate how useful the oracle is in solving problems, while the latter reveals conditions where, even if an oracle was unable to settle a tie, it was nonetheless able to express preferences similarly to humans.

Our distance measure is derived from Jensen-Shannon Divergence (JSD), which estimates the similarity between probability distributions by measuring mutual information [98]. We use the Jensen-Shannon distance, a metric form of JSD found by taking its square root [31]:

$$JS(O, H) = \sqrt{\frac{1}{2}\left(D_{\mathrm{KL}}(O \parallel M) + D_{\mathrm{KL}}(H \parallel M)\right)}$$

where O and H are the oracular and human answer distributions for a single problem, M is their pointwise mean, and D_{KL} is the Kullback–Leibler divergence. By finding the mean JS distance between oracles and humans, across all problems in the human set, we estimate their similarity.

Inspecting the results in Table 5.7, we find that the measures are not correlated with each other, particularly when comparing *Likelihood* and *Stack* oracles. While the likelihood is more predictive of the decisions humans actually made, it was more idiosyncratic, having an overall higher JS distance compared to the rule-stacking oracle. While the prior more accurately captured the human answer distribution, it is too coarse-grained to be able to break subtle ties. Performance of all oracles is shown to progressively degrade as problems become more ambiguous, requiring more and more sources of knowledge with which to base decisions upon. Encouragingly, the Bayesian oracle is most closely matched to human performance in both accuracy and distribution, across all problem types.

5.6.4 Prioritisation and induction

Bayesian-RAVEN relaxes the well-structured nature of RAVEN problems, forcing solvers to re-contextualise knowledge and appraise additional sources of information. In doing so, it aims to disqualify a classification strategy, in which solvers pick answers based on the presence of surface-level statistics, instead of inhabiting a deeper generative model with which to rank hypotheses. The most challenging split is therefore Induction-Plus, as it trains on all problem types except for *New information*, and introduces *Induction* problems (requiring integration of prior and likelihood) at test time. This experiment is therefore expected to exacerbate several shortcomings of deep-learnt baselines, including their data-inefficiency and incapability to learn generalisable features "on their own", when not forced to by the training set itself. From what we have learned in this thesis so far, such models must

Split	All		Basic		Break		Priorit.		Induct.	
Measure	#	*JS*	#	*JS*	#	*JS*	#	*JS*	#	*JS*
Oracles										
Stack	11	0.0981	11	0.0764	0	0.1128	0	0.1098	0	0.1476
Prior	16	0.0970	11	0.0751	5	0.1120	1	0.1152	0	0.1510
Likelihood	24	0.1022	11	0.0794	12	0.1177	9	0.1125	0	0.2056
Bayes	**25**	**0.0946**	11	**0.0741**	**14**	**0.1086**	**10**	**0.1032**	**1**	**0.0974**
Total problems	32		13		19		14		1	

TABLE 5.7: A total of 32 problems were shown to all human participants, with each problem belonging to at least one split. Table entries communicate the number of problems where each oracle correctly predicted the mode of human responses, as well as the Jensen-Shannon distance between oracle outputs and human responses averaged over each split. The Bayesian solver is shown to most closely match human performance in both accuracy and distribution.

be given an impetus to find these concepts — or alternatively, denied other paths — or they will take shortcuts. Core to Bayesian-RAVEN is the stance that, if we do not rectify this by imbuing our models with a more general inductive bias, they will remain brittle to the kinds of distributional shifts that humans adeptly cross.

Looking to Table 5.8, we find that *Rel-Base* outperforms both *Stack* and *Prior* oracles when allowed to train on the full Prioritisation set. From here, performance begins to decline as test problems focus on induction alone. Interestingly, for this set, *Stack* remains around 50%, meaning that a solver able to count rules is still likely to get half of the problems correct. However, Rel-Base declines to 44.60%, again suggesting that it struggles with distributional shifts. We believe this validates the direction of Bayesian-RAVEN, and indicates that these splits are imposing a significant challenge. Additionally, training on 25% data (equivalent to 3,000 training problems) cuts performance significantly for Rel-Base, lowering to near-random performance overall.

Observing blind performance, we achieve similar results to those seen earlier, with *Independent Answer* and *No Context* blinds resembling random chance. *Last Row* has also diminished, never achieving more than 38.58%. These results confirm that there are no obvious sources of exploitation introduced by generating and partitioning problems in this way.

Oracles reveal a fairly even balance on the Induction task, all achieving around 53%. This is to be expected, as these problems are defined as being ambiguous to prior and likelihood alone; solving these problems while lacking one or more pieces of information will result in encountering ties resolved by chance. The fact that *Stack* performs at around 50% tells us that most induction problems involve a two-way tie. Meanwhile, on the Prioritisation split, *Likelihood* scores far higher than *Prior*, which is also to be expected. Since the likelihood function is more expressive and fine-grained than the prior distribution — involving more knowledge derived from the

Split	Prioritisation			Induction			Induction-Plus		
Data	*10%*	*25%*	*100%*	*10%*	*25%*	*100%*	*10%*	*25%*	*100%*
Solvers									
Rel-Base	31.05	33.22	**57.90**	25.93	26.20	**44.60**	25.58	26.24	**36.43**
Blinds									
Indep. Ans.	27.00	26.60	27.00	24.30	27.33	25.45	25.47	25.12	27.17
No Context	24.82	26.92	26.28	24.90	21.84	23.99	27.23	20.89	26.51
Last Row	28.85	30.62	38.58	31.91	25.25	26.09	22.40	27.23	25.68
Oracles									
Stack		45.43			51.09			-	
Prior		54.88			53.01			-	
Likelihood		71.83			55.60			-	

TABLE 5.8: Induction performance of solvers, blinds, and oracles, for varying amounts of training data, as compared to decisions made by the Bayesian solver (% accuracy). Oracle results aren't recorded for Induction-Plus; as oracles aren't trained, the task is identical to Induction.

data such as frequency of hypotheses, as well as contextual information in the form of similarity scores — this means that the prior will encounter less problems that it can successfully disambiguate, and be at the whims of `argmax` picking between identically-ranked candidates.

5.6.5 Generalisation

As problems in this dataset are labelled with both answer targets and probability distributions, we experiment to see whether the choice of training signal improves data efficiency, aiding generalisation performance. Conceptually, if we believe knowledge of a solver's "runner up" choices to present a source of useful information in testing, then we should also expect this to be informative to a model during training. Table 5.9 presents confirmation of this, with the most visible benefits — of training to predict the labelled distribution — being seen when Rel-Base is data-limited to 25%. This is met with a rise in performance from 34.78% to 40.62%. Overall, performance is still quite low, with the best performance achieving just above 50%. Therefore, we believe these problems are encouraging of a new wave of work in the RAVEN area.

5.6.6 Brittleness

Recall from Section 5.5.2 that we "trained" humans on high-certainty problems by exposing them to ten, pre-solved examples before their test phase. Although not directly comparable since humans leverage much more knowledge than is formed during their trial, here, we try to run a similar test of brittleness by training baselines on Neutral-HC (*Neutral* problems with high certainty s), and testing their performance as problems become less certain (δ_y, as measured by the Bayesian oracle).

Split	Generalisation					
Target		*Single*			*Distribution*	
Data	*10%*	*25%*	*100%*	*10%*	*25%*	*100%*
Solvers						
Rel-Base	31.50	34.78	48.38	33.65	40.62	**52.12**
Blinds						
Indep. Ans.	25.62	26.22	**27.60**	26.82	26.95	25.80
No Context	24.22	25.32	**26.88**	26.25	25.25	26.65
Last Row	29.20	31.42	35.15	29.95	30.55	**37.30**

TABLE 5.9: Generalisation performance of machine learning baselines, investigating the effects of changing both training signal and quantity of data (% accuracy). Note that results are emboldened by row, to indicate best settings per model.

Split	Neutral-HC	Brittleness	Brittleness+
Solvers			
Rel-Base	81.92	74.08	53.38
Human	75.00	87.50	62.50
Blinds			
Indep. Ans.	28.00	26.18	25.12
No Context	40.48	25.68	26.18
Last Row	60.55	47.32	39.28

TABLE 5.10: Results from the brittleness experiment comparing Rel-Base to human performance (% accuracy).

Table 5.10 reports performance degradation on all solvers as test problems become more ambiguous when compared with training problems, as expected. However, human performance remains more consistent than Rel-Base.

Here, we encounter the limitations of the scale of our human study, as the reported performance on the Brittleness split is higher for humans than Neutral-HC. We calculated that this is the difference of one problem; when there are only 32 problems being solved by all participants, only 8 problems per these three splits were able to be observed. Nonetheless, we take this as an indication of graceful degradation, and leave larger-scale and/or targeted human experiments to future work.

Regarding blinds, we see the accuracy of *Last row* creep up to 60.55%, which is comparable to its performance on the Basic split. As commented earlier, Basic and Neutral-HC splits are expected to be the easiest (and most exploitable) because they feature the least ambiguous problems, leaving enough information present in the last context row (especially when there is not a `Distribute-Three` rule present in the problem). As before, we believe these results are consistent with the format being sound enough to facilitate experimentation, as long as the blinds have first revealed a lower bound of what is achievable.

Confidence		Formal measure		
		Low	*Med*	*High*
Human	*Low*	9	0	0
	Med	8	5	1
	High	2	1	6
κ: 0.5234				

TABLE 5.11: Confusion matrix showing agreement between human confidence (self-reported) and a formal measure derived from the Bayesian oracle. Cohen's kappa is also provided.

5.6.7 Confidence and inter-rater reliability

For the final experiment in this chapter, we aim to determine how well our formal measure of confidence (recall $\delta_y \omega_y$, Section 5.4.3) correlates with confidence scores as rated by humans. Because our measure of confidence is continuous, we take the confidence values (as assigned by the oracle) of the set of problems shown to participants, normalise, and quantise to [1..3]. This allows direct comparison with the Likert scale used in the human experiment. Normalisation to [0,1] is achieved by feature scaling: $\frac{x-\min(x)}{\max(x)-\min(x)}$, where x is the set of confidence values. Quantisation then bins values with a step size of $1/3$.

To visualise our results, Table 5.11 presents a confusion matrix showing how problems were labelled by both humans and the Bayesian oracle. At a glance, we see a shaded diagonal that points to some correlation. Areas of difference are mainly due to the formal measure underestimating confidence, labelling 8 problems (25% of the human set) as "low", when humans instead labelled them as "medium".

To further quantify this correlation, we calculate Weighted Cohen's Kappa as a measure of inter-rater reliability [26], with linear weights. This is a suitable metric given our samples are ordinal, and the assumption of rater independence holds. Our resulting kappa of 0.5234 indicates moderate agreement, which is a promising early result in this direction, but indicates that future experiments using this measure of confidence will benefit from fine-tuning.

5.6.8 Success and failure cases in the human set

To elucidate further differences between solvers, Figures 5.5, 5.6, and 5.7 display the output of both human participants and our Bayesian model, on eight problems randomly selected from the dataset (four successes, four failures). These problems are also selected to represent *basic*, *tie-break*, and *new information* problem types. For each problem, we try to localise points of difference. While we mark a failure case as occurring when the human vote did not match the Bayesian model, we suggest that "failure" in this sense is to be taken with an asterisk, as there are problems in this section that are — to the author's mind — 'incorrectly solved by most participants. Despite the subjectivity incurred by such a dataset, these comparisons are still quite useful, as it *seems* that humans display more certainty than the Bayesian solver

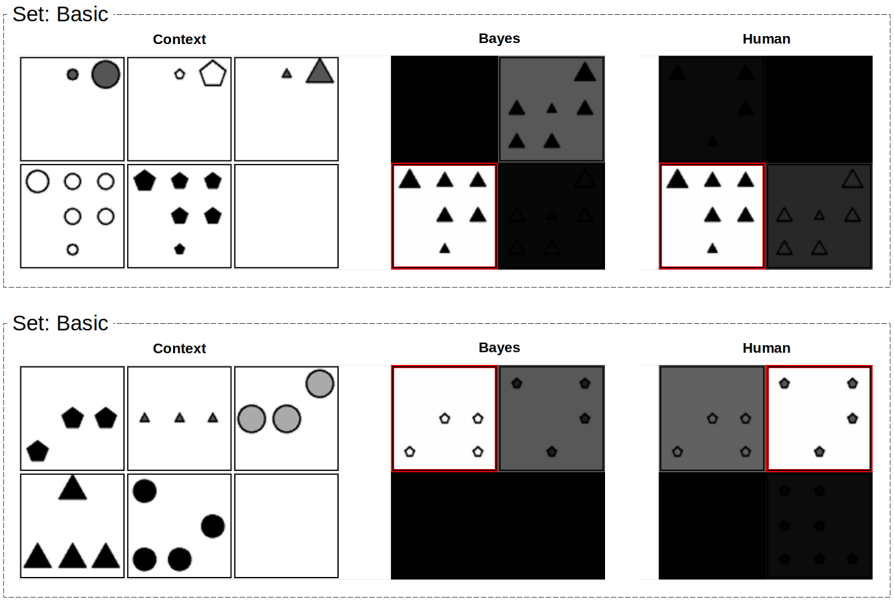


FIGURE 5.5: Two problems from the *basic* set.

overall (which can be surprising, given the level of ambiguity). It is unclear whether or not this is truly the case, given the limitations of the study format (in particular, asking each participant for a single response instead of a probability distribution). Figure 5.5 presents two problems from the *basic* set. In the upper problem, we see that the Bayesian solver has successfully predicted the human answer, although, it has calculated that the other frame consisting of dark triangles (answer two) is also of slight consideration. While participants largely selected answer three, we expect that they found the white triangles of answer four to enable a foil rule not found in the dataset. Hypothetical rules operating on colour, including any of `Odd-one-out`, `Distribute-Two`, `Middle-Different`, or `Alternation`, would fit this sequence. The Bayesian solver could only work with the rules it knew, and therefore did not rate answer 4 more highly. Looking to the bottom problem, we see a failure case involving the Bayesian solver choosing the `Arithmetic-Colour` rule over the interval rule (i.e. all similar colours). While we think the former rule is ultimately more correct, it is also unintuitive and hard to perceive in this format.

In Figure 5.6, we see four problems from the *tie-break* set. The top problem reveals a successful inference made by the Bayesian solver, as seeing both rows contain a permutation of one, two, and six shapes was slightly more surprising than matching colours. For the second problem, we compare the output of the likelihood alone, instead of the Bayesian model, to illustrate that it was sufficient to predict the human response. However, this is a failure case, as the prior's stronger preference for `Distribute-Three-Type` over `Progression-Number` tipped the Bayesian model towards answer three, consisting of three white circles. The third problem reveals another success, as the solver predicted the shape type to be more important than the

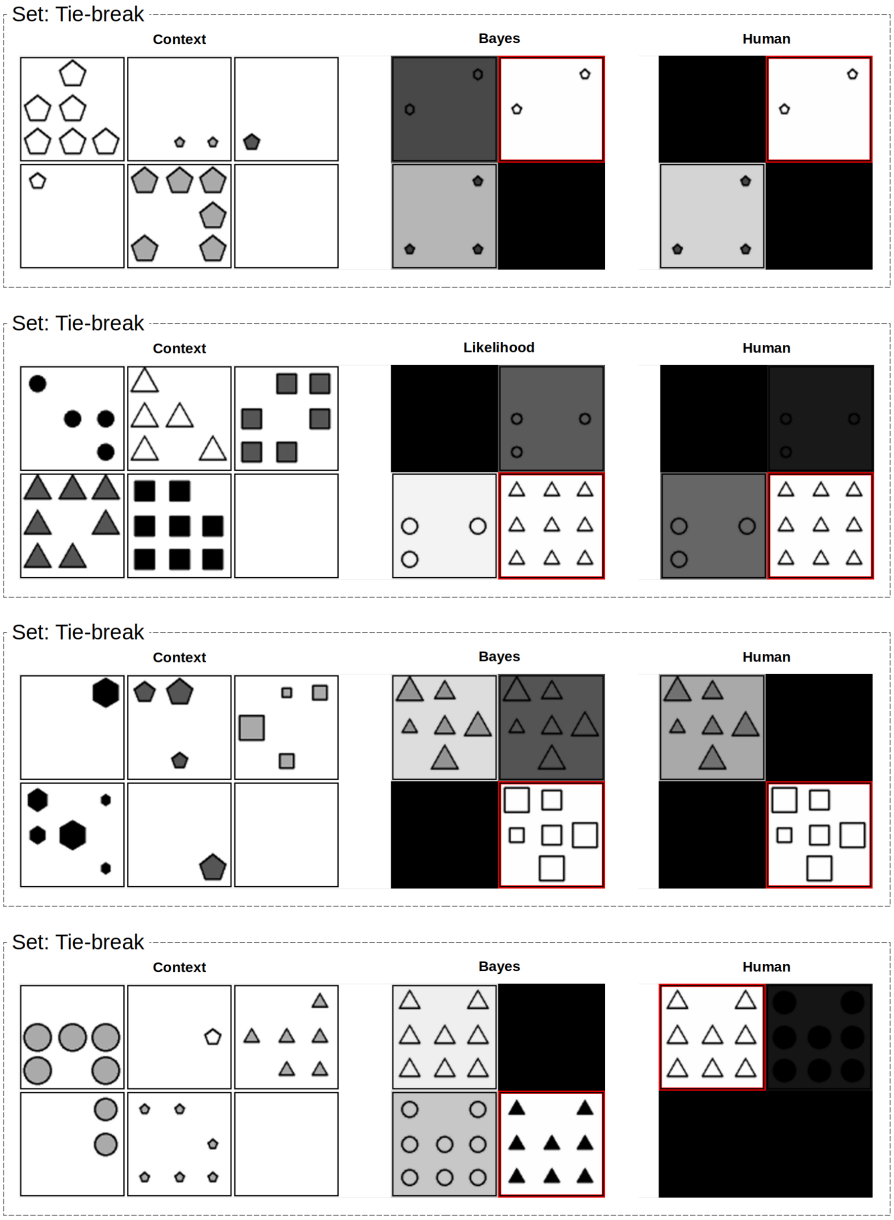


FIGURE 5.6: Four problems from the *tie-break* set.

colour progression in this context. Finally, the last problem displays a very incorrect assessment by the solver. For humans, we again expect that the `Arithmetic-Colour` rule went unseen, due to rules not present in RAVEN (e.g. `Distribute-Two`) becoming apparent. Given the colours in the context, we expect this new rule combined with an interval rule (e.g. `Interval-Colour-Light`), rendering the presence of dark shapes highly irregular.

In Figure 5.7, we see two last cases, one success and one failure, from the *prioritisation* set. Key to understanding this failure case is that the likelihood function modifies its hypothesis scores given the properties of the incomplete row. With all else being equal, and no mathematical rules found to explain the number or position of shapes,

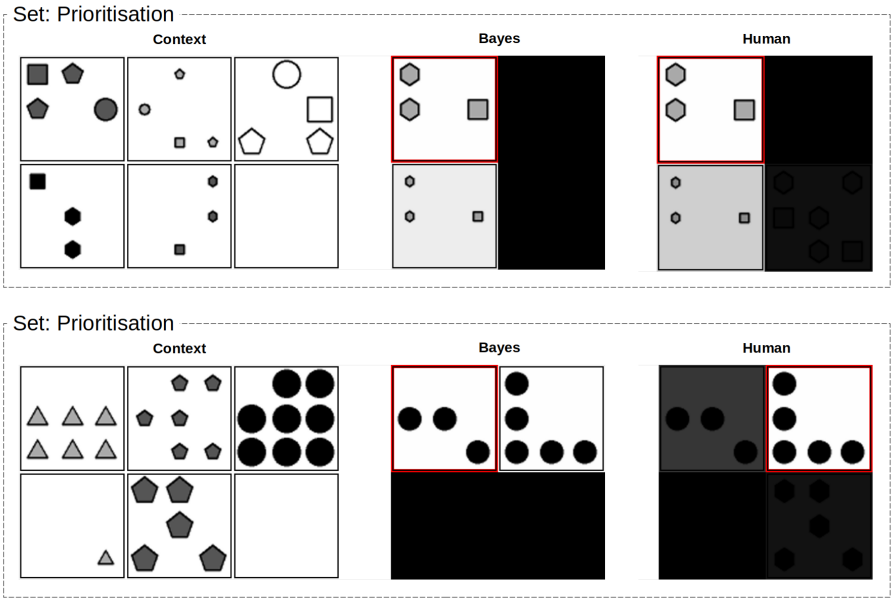


FIGURE 5.7: Two problems from the *prioritisation* set (i.e. *new information* problems, to recall the problem types listed in Section 5.5.1).

it considered the first answer to at least be closer to the average number of shapes for that row. However, we expect that humans recognised a different interval rule, for example, `Interval-Number-High` (this *may* read as: *"each frame should be more than half full, except for that fourth frame, which I am deliberately ignoring"*). While neither humans nor the Bayesian solver had a more compelling rule to make use of, we think that participants resolved this ambiguity better than the solver, as they chose the frame that had a number of shapes closer to the context's average.

5.7 Limitations and future work

Limitations of human data

PMPs are intuitively presented to humans as selection tasks, where one has to pick the most suitable answer from a set. These problems becomes cumbersome if one is asked to rate each answer independently, or to assign a probability distribution to the set of answers directly. Our experimental design was to therefore ask all participants to solve the same set of 32 problems, in order to obtain a probability distribution over the answers of each problem, at the cost of testing more broadly across the dataset. In Section 5.5.2, we mentioned that the only requirement for sampling test problems for the human set, was that each of the problem types needed to be represented. We consider it to be improper judgement in hindsight that we did not balance this perfectly, as the rarity of induction problems meant us being unable to properly benchmark human responses for that problem type. While we draw attention to these limitations in the chapter, it should remain considered when

interpreting our results (particularly when comparing human and machine learning models), and lead on to further work involving more targeted human experiments. Our goals in this chapter were broad, seeking first to direct the methodology in this area and establish the Bayesian-RAVEN dataset as a response. We expect that there are many elements of our investigation capable of receiving their own in-depth treatment down the line.

Modelling human performance

In presenting ambiguous problems, we have also introduced a degree of subjectivity. This means that there might not exist a 'true' correct answer to these problems, even among human peers, as tie-breaking will be contingent on both one's generative model and priors. As stated earlier, our goal was not to obtain a Bayesian model that precisely fits human perception, but serves to fulfil a set of experiments not otherwise attempted with the RAVEN set, investigating statistical models through a Bayesian lens. Therefore, we also see potential for future work in cognitive science. We expect that our extended consideration — of the ways in which we might make more discriminative PMPs for machines, given the different forms of inference — will be of use to all applications of PMP research. Ideas in this chapter have the ability to inform PMP design for human psychometrics, potentially leading to the creation of another advanced matrices booklet that could correlate quite well with high IQ.

Acquiring priors

Constructing a prior in the manner detailed in Section 5.4.1 does not escape the problem of subjectivity. There are infinite ways we might have selected a set of fundamental operations, programs, and complexity measures. Solomonoff probabililty is uncomputable, and laying mathematically-rigorous foundations for useful approximations is a rabbit-hole that is thoroughly out of the scope of this thesis. Our hope is that our motivation of this approach, as early work in Bayesian modelling for RAVEN that also incorporates inference to the best explanation, may inspire other researchers to do this justice.

That being said, we tip future work with a suggestion involving algebraic machine reasoning. In [170], concepts in RAVEN are modelled as ideals in polynomial rings, while patterns involving these concepts are induced via primary decompositions of ideals. To join this with our present work, the hypothesis space might be populated by the discovered monomial ideals, with the prior being generated automatically by using the height of ideals as an estimate of algebraic complexity. We are excited by the prospect of future solvers possessing the ability to extract their own Bayesian components (hypotheses, priors, and likelihoods) in ways that demonstrate more universal inductive reasoning.

Compression

In future work, a related way to view the RAVEN task would be through the lens of compression. In these terms, when we say the job of a solver is to pick the answer that maximises regularity, we mean that it finds an answer to minimise the encoding length of the completed problem, i.e. maximising compression. Now, let us assume that a solver utilises a compression program in the following way: it abducts a hypothesis h, it passes both problem and h to the compressor as strings, and the compressor returns the compression ratio achieved when conditioned on h. The length of the compression program, as modulated by h, can be thought of as corresponding to the complexity of h, and therefore a proxy for a Bayesian prior. Computing the resultant compression ratio produced by the program under h resembles a Bayesian likelihood function.

To rebuild our Bayesian oracle in a way that is amenable to end-to-end deep learning, we could train a language model as a compressor. The connection between language models and compression — notably, Transformers — is well established [28]. Perplexity, a ubiquitous measure of the performance of language models, is itself an exponentiated compression ratio.[6] We imagine that, by first putting a language model into a particular state by passing it tokens representing hypothesis h, then measuring the perplexity of a completed PMP problem to follow, we have also been told something about the compressibility of that problem, given a) the answer frame selected, and b) the model in state h. Putting these ideas together, a solver could make use of the raw patterns discovered by an algebraic machine reasoning framework [170] to condition a Transformer, learning to abduce hypotheses based on the compressive utility of each of these patterns, towards the evaluation of PMP answers. As Bayesian-RAVEN has labelled all problems with a string representation, we hope to enable future work in this exciting space.

Broadening our measure of problem quality

As the different problem types can be placed on a gradient of increasing ambiguity — that is, *basic* → *tie-break* → *new information* → *induction* — there exists a trade-off between the formedness of a problem and its discriminative utility. Not enough ambiguity, and the problem does not produce enough data for investigators looking to discriminate between solvers; the answer is either correct, or incorrect. Too much ambiguity, and the problem ceases to be well-formed, enough for there to be a clear answer at all. We imagine that "Goldilocks" problems will provide just enough ambiguity to be highly challenging, while remaining able to be reliably solved by "rational" agents. While our work has taken the first steps in introducing this to the deep learning literature, there is still much to be explored in striking this trade-off just so.

[6] Marcus Hutter makes this point on his personal website:
http://prize.hutter1.net/hfaq.htm#perplexity

Picking up the thread from our efforts to formalise a measure of problem confidence, there is potential to develop this towards optimising this trade-off, applying Bayes to problem selection. We also see a more general application of this to puzzle design. David Wilson, in his "Introduction to Writing Good Puzzle Hunt Puzzles", similarly refers to a notion of *elegance*, although informal: "[elegance] is generally a feeling that the puzzle is a coherent whole".[7] They go on to flesh out elegance as suggesting that a puzzle "wastes nothing... every clue contributes to the final answer", and that if a puzzle contains multiple levels, "it is elegant to repeat the same operation at each level". The former point bears great resemblance to Occam, and we interpret this to mean that puzzle design should reward inference to the best explanation, and not by multiplying entities beyond necessity. The latter point speaks to compression: if the application of a single hypothesis allows us to compress a puzzle, requiring very little in the way of other information to take us from beginning to end, then that hypothesis has high explanatory power, and the solution of our puzzle should be met with a satisfying *eureka!* A good puzzle should not leave a person thinking "how was I supposed to guess that?" We can also see this in action in the Stroop effect [154]; when stimuli are conceptually congruous, they prime activation of related concepts that can guide us to a correct perceptual "take", much faster than if they were incongruous. For example, writing the text, "upside down", *upside down*, allows for what might be initially apparent — flipped text — to cue something else about the stimulus.

5.8 Conclusion

This thesis asks: *to what extent can we claim our systems are imitating the kinds of reasoning we want to capture? How do we prepare models to perceive scenes at the correct levels of abstraction, attending to features that are justifiable given context?* In this chapter, we have pursued a theoretical answer from Bayes; the correct level is described by the leading hypothesis, considering both elegance and explanatory power. Applying this to RPMs, we have an experimental answer: we created Bayesian-RAVEN, a derivative of the RAVEN set, with answers chosen by a Bayesian oracle. This new dataset has allowed us to further probe the induction and generalisation abilities of our models, and their similarities to humans, over several key experiments.

We established that knowledge of a solver's "runner up" choices present a previously untapped source of information in our area, allowing us to invite a degree of diagnostic ambiguity to the problem format. In pursuing more discriminative problems, our theory has implications for high-IQ human psychometrics as well. We also maintained the use of blind solvers throughout, to ensure that modifications to the problem format did not re-introduce shortcuts.

General performance experiments revealed that while our machine solver begins by matching human performance on basic problems, it declines much faster

[7] Wilson's introduction: mit.edu/ dwilson/puzzles/puzzlewriting.html

when asked to navigate ambiguity, which is not revealed by testing on problems in RAVEN. Our Bayesian oracle is also shown to be quite predictive of human responses, both in their chosen answers, as well as their self-reported confidence, potentially leading to further work in cognitive science and automated problem design. Finally, we leave the community with a discussion of many worthwhile directions for future work, including thoughts on how to more fully integrate the Bayesian framework with deep learning approaches, and minimise the design areas that involve subjectivity, such as hand-constructed priors.

Together with the previous chapter, this work takes a step towards engineering systems in a way that is far more intentional about *what* reasoning abilities are desired, and *how* we might understand their existence. We endeavour to reinstate the raison d'etrê of PMP research in our area — measuring abstract reasoning in neural networks, without mistaking shortcut learning for skillfulness.

Chapter 6

Evolving PMPs with Unicode Analogies: An Anti-Objectivist Visual Reasoning Challenge

6.1 Preface

In Chapter 3, we confirmed that — as in other applications of deep learning — our models will uncover exploits in our tasks that we do not realise are there. In Chapters 4 and 5, we argued the importance of leaving models "no way out", other than to tackle the intended tasks head-on. We offered a space of problems and valid solution strategies, pointing to a type of problem that may be harder to shortcut due to the requirement of reappraising rules in new contexts, instead of simply recognising their existence. This culminated in offering *Bayesian-RAVEN*, an extension to RAVEN that reduces the success of solvers that memorise rules. In this chapter, we repeat this trick — not on rules, but on the attributes themselves — dismantling the hard-and-fast perceptual boundaries of the RAVEN set entirely. The language used by this chapter frames this as "anti-objectivism", in recognition of Bongard's own Gestalt leanings and pursuit of problems designed to evade algorithmic approaches.

Recall the polysemic image in Figure 6.1, which we first introduced in Chapter 2. As a matter of fact, this figure is character U+1FBBE from the *Unicode standard*. In the dataset introduced by this chapter, it is also used in problems that play with low and high ink level, in problems that involve triangles, in problems that explore diagonally-arranged objects, the number of solid components, an equal aspect ratio, a centred mass, two base contacts, flat base contacts... and the list goes on. We reiterate that settling on a perceptual take is therefore a question of context and of usefulness, and not of assigning an objective label. Our dataset, *Unicode Analogies*, directs PMP research in the direction of Bongard, asking solvers to co-refine both observations and hypotheses, requiring more of a cycle of inference (such as the one presented earlier in Figure 4.3).

This chapter consists of our article as it appeared in the Proceedings of the *Conference on Computer Vision and Pattern Recognition* (CVPR) 2023, in a single-column format for stylistic consistency, and altered to remove self-citations.

FIGURE 6.1: Unicode character U+1FBBE

6.2 Abstract

Analogical reasoning enables agents to extract relevant information from scenes, and efficiently navigate them in familiar ways. While progressive-matrix problems (PMPs) are becoming popular for the development and evaluation of analogical reasoning in computer vision, we argue that the dominant methodology in this area struggles to expose the lack of meaningful generalisation in solvers, and reinforces an objectivist stance on perception – that objects can only be seen one way – which we believe to be counter-productive. In this chapter, we introduce the Unicode Analogies challenge, consisting of polysemic, character-based PMPs to benchmark fluid conceptualisation ability in vision systems. Writing systems have evolved characters at multiple levels of abstraction, from iconic through to symbolic representations, producing both visually interrelated yet exceptionally diverse images when compared to those exhibited by existing PMP datasets. Our framework has been designed to challenge models by presenting tasks much harder to complete without robust feature extraction, while remaining largely solvable by human participants. We therefore argue that Unicode Analogies elegantly captures and tests for a facet of human visual reasoning that is severely lacking in current-generation AI.

6.3 Introduction

Traditionally, statistical classification models have been designed to neatly cleave data into categories. Even in tasks such as visual scene decomposition, where data resists full description by any one label, there is an underlying objectivist assumption being made; the expectation of there being an objective number of distinguishable "things" present, themselves belonging to singular classes. Human visual perception makes a departure from this. The symbolic world to which we attend, with firm compositional rules for scenes and their objects, and with their parts and positions, is subsisted by a churning sea of ongoing conceptualisation processes deeply fluid and contextual [69].

In recent years, there has been a proliferation of computer vision architectures

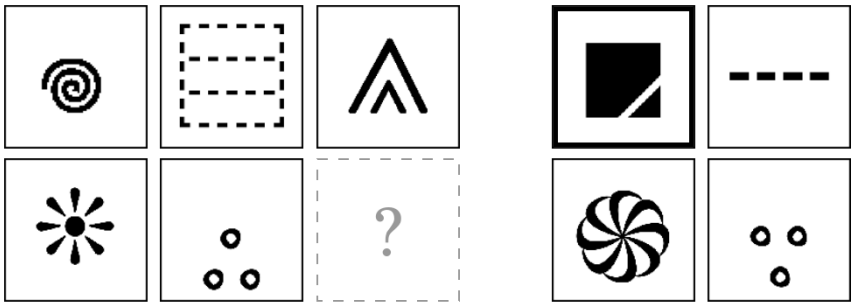


FIGURE 6.2: An example problem in UA, instantiating the *Distribute-Three* rule with the *Closure* concept. Five out of six context frames are provided (left), with four answer frames to choose from (right). The correct answer is emboldened.

built with object-centric inductive biases [101], many of which represent states-of-the-art on popular datasets [176, 32]. This is an important direction, as training models to decompose scenes into objects allows for an explicit abstraction stage promoting feature reuse. However, abstract visual reasoning tasks such as Bongard problems [11] expose philosophically [99] — and in this chapter, experimentally — that such an approach might work against the creation of models that possess the ability to abstract and deploy useful concepts. This observation also engages a current debate in the literature regarding the scalability of built-in knowledge and inductive biases [105, 155].

Humans display flexibility in how they decompose scenes, and perceive such scenes at a level of abstraction informed by past experiences and appropriate to present goals [42, 43]. Scene understanding in humans is therefore undergirded by something other than the perception of static objects [103], and the idea that scene modelling research can separate perception and higher cognition into a pipeline of self-contained modules is strongly critiqued [21].

Noticing other shortcomings of deep-learnt approaches to computer vision, including brittleness to out-of-distribution (OOD) data, a small number of abstraction datasets inspired by Raven's Progressive Matrices have been recently released [104]. Further motivations to this direction include a) the expectation that tasks with such an extended history in general psychometric testing would be useful to import into computer vision research, and b) the opinion that the more broadly applicable a model's abstracted concepts become, the more robust that model will be under OOD conditions [120]. While the applicability of such concepts should ideally be evaluated by these datasets, common approaches to dataset creation feature conceptual schemas consisting of simple objects that can be neatly dropped into scenes, and extracted by scene decomposition stages (such as the one present in Rel-AIR, introduced in Section 3.6.3). This seems to require little in the way of contextual perception, such as Hofstadter's notion of "conceptual slippage" [70].

We observe that the world's writing systems present a diverse resource of characters that are amenable to content analysis, and can assemble novel reasoning problems of their own. We introduce the *Unicode Analogies* (UA) challenge, consisting of character-based progressive matrix problems (PMPs) to benchmark fluid conceptualisation ability in vision systems. The characters in UA are polysemic, and may instantiate any number of concepts, with the salient concept only revealing itself given context (Figure 6.2). By generating training and testing problems from disjoint sets of characters, we challenge these systems by presenting tasks much harder to complete without robust feature extraction, while remaining largely solvable by human participants. In doing so, we contribute a dataset that unlike others in this area, operates on a rich conceptual schema that invites fine-grained experimentation, and is easily extensible to new user-defined concepts. Over five key experiments, we explore human and model performance on a number of dataset splits generated by UA, demonstrate that state-of-the-art solvers are still far from achieving the founding goals their datasets were created for, and encourage new solvers to overcome these limitations.

6.4 Background

6.4.1 Vision vs. objectivism

For humans, our perceptual world is not populated by firm and unchanging concepts, as if there were some neatly defined mental collection. There is a wealth of psychological research to suggest that cognition – at the levels of conceptualisation [18], reasoning [48], and memory [131] – operates on concepts that are blurred, evolving, fluid, and *ad hoc*. Consider the child who perceives a tree stump surrounded by mushrooms as a dining setting for small creatures. Such analogies are ubiquitious in how we understand scenes not simply as lists of objects, but microworlds with physics, rules, structure, intent, and purpose. Via analogy, these worlds, which may not be previously experienced, are "seen as" familiar, in order for us to successfully traverse and manipulate them, efficiently guiding perception and problem solving [51]. We believe that a deep understanding of concepts is demonstrated by the ability to both perceive them in diverse stimuli, and to leverage them for utility. Echoing Odouard and Mitchell [120], the way we assess trained models needs to remain fully aware of this.

6.4.2 Progressive matrix problems and deep learning

Since their introduction in 1936, Raven's Progressive Matrices (RPMs) have seen extensive use in psychometric testing [128, 127], in part due to their abstract, non-verbal, and assumedly culture-agnostic design, as well as their simplicity to administer. RPMs present a visual pattern-matching task requiring solvers to perform analogical reasoning, and such reasoning must depend on the company of context images if a solution is to be found and analogy drawn.

In the field of computer vision, deep learning is ubiquitous in leading models, bringing with it both the remarkable ability to perform rich, automatic feature extraction from large datasets, and a severe brittleness to out-of-distribution data. As analogical reasoning in human and non-human animals is hypothesised to support feats such as tool use and creation, and indeed, general problem solving [69], there has been much interest in creating RPM-inspired datasets amenable to deep learning. In this chapter, we refer to the problems presented by all such datasets as belonging to the class of progressive matrix problems (PMPs).

The last five years has seen the release of several abstract reasoning datasets, including two seminal PMP datasets; PGM [4] and RAVEN [174]. PGM is considered the first large-scale dataset of its kind, while RAVEN builds upon it, increasing the diversity of rules and configurations instantiated by problems to discourage memorisation and more accurately assess the generalisation ability of trained models. Since RAVEN's release, there have been a number of research efforts (reviewed by Malkinski and Mandziuk [104]) to benchmark novel architectures on its problems, each analysing model performance primarily informed by overall accuracy. While this presents as a fairly standard methodology in machine learning research, there are more nuanced considerations that this branch of research demands.

6.4.3 Shortcuts and non-robust features

For any given data, there exists a landscape of "perfect" models, i.e. those that have full explanatory power for those data. Knowing which models will also ultimately capture the knowledge to describe additional data is a contentious question [105]. Humans have evolved many biases, such as the preference for simple explanations [161]. Ironically, the tendency to use analogies has meant expecting broader cognitive abilities of our seemingly mind-like models.

Recent works have exposed the existence of shortcut and non-robust feature learning in neural networks [46, 90]. Geirhos et. al communicate that shortcut learning is a failure to generalise "in the right direction", where a model extracts and depends on features that are not present OOD [46]. Similarly, Langosco et. al explain that learning non-robust features and objectives occurs when a network encapsulates the "wrong" knowledge, i.e. that fulfils optimisation in a way that wasn't intended by researchers [90].

In our research area, such phenomena have resulted in networks failing to learn generalisable features, and exploiting biases in PMP answer sets without the awareness of researchers at the time of publication. Recognising this as an important consideration for dataset creation, we have designed splits in Unicode Analogies to assemble train and test problems from disjoint sets of images, requiring models to learn robust features if they are to perform well.

6.4.4 Comparisons to other datasets

Datasets such as PGM and RAVEN represent important developments in this field, being the first to automate PMP generation at-scale for deep learning, and with enough diversity to pose a challenge for machine solvers at their times of release. However, they also represent one particular approach to PMP formation, adopting basic object-based schemas, and building complexity by stacking multiple rule instantiations in a given problem. While more recent architectures have become adept at modelling the default splits of these datasets, there is less focus on universally poor extrapolation performance, which has seen relatively little progress [104]. To account for this discrepancy, we hypothesise that this approach to PMP generation and testing is not fully diagnostic of an architecture's analogical reasoning abilities. The familiarity of stimuli invites architectures to separate perception from higher-order cognition, allowing much of the work of the problem to fall to representation learning. If it were not so, rule-stacking would have a greater negative impact on performance than is observed, and introducing modified stimuli would be less detrimental.

Unicode Analogies (UA) is able to broadly express the schemas utilised by these datasets, including a familiar exploration of rules such as progression and arithmetic, objects including shapes and lines, and attributes like size and number, to name a few. However, these are situated within a far richer schema of concepts at multiple levels of abstraction, many of which are inspired by Bongard problems [11] and principles of gestalt perception, including closure, negative space, and grouping. In doing so, it blurs the lines between object and feature, and between perception and cognition, forcing models to incorporate contextual information at all stages of problem solving. This dataset brings PMP research in-line with philosophical criticisms of objectivist approaches to AI [21], prohibiting solvers from relying on scene decomposition stages. It presents just one rule per problem, asking solvers to discover what is salient, instead of learning to represent scenes *a priori*. It also responds to the call for datasets to support concept-based evaluation as voiced by Odouard and Mitchell [120], which is a valid criticism across all other datasets we are aware of.

Most similar to our work is the Bongard-LOGO dataset [117], which also motivates context-dependent perception as a crucial property of human cognition. Bongard-LOGO presents a few-shot benchmark intended for meta-learning, and focuses on capturing the Bongard problem format with frames consisting of generated line

drawings. UA instead benchmarks supervised learning approaches such as those built for RAVEN and PGM, importing concepts from Bongard and other formats into progressive matrices. Bongard-LOGO exclusively investigates invariant perception with regards to size, orientation, and position, whereas Unicode Analogies does not limit itself in this way.

Other notable datasets, including KANDINSKY [72], ARC [25], PQA [126], and LABC [65], have related goals. The KANDINSKY set explores spatial and gestalt visual tasks within the context of explainability research, and suggests importing such concepts into progressive matrices as future work. ARC presents a corpus of hand-designed intelligence tasks that require models to generate coloured grids. ARC contains a broader, more complex task base than UA, and is intended to be a very general battery for machine intelligence testing, whereas UA is focused primarily on fluid perception. PQA borrows ARC's gridworld format, and introduces seven tasks related to laws of gestalt perception, already largely solvable by the technique offered in the paper. LABC offers a two-row variant to the problems in PGM, focusing on analogical reasoning across domain shifts, while inheriting the same basic schema as PGM.

6.5 The *Unicode Analogies* framework

Unicode Analogies is an extensible framework that allows for the creation of character-based PMP datasets from a conceptual schema. In this section, we introduce this work as a pipeline from schema formation, character annotation, and problem generation, through to defining training splits.

6.5.1 Defining an expressive conceptual schema

Starting with known concepts with historical usage in PMPs and Bongard problems (e.g. rules involving progression or distribution, and features such as size and shape), the first author performed content analysis on hundreds of characters appearing in the Unicode standard. By following a conventional approach to content analysis resembling Mayring's inductive category development [107], we formalised a broad conceptual schema with which to annotate more characters, allowing software to assemble thousands of novel PMPs. The current conceptual schema is shown in Figure 6.3, and depicts many concepts across multiple levels of hierarchy. Generated problems express one of 5 rule types: *constant*, *progression*, *arithmetic*, *distribute-three*, and *union*, with each rule applicable to a subset of concepts. The first 4 of these rules are explored in RAVEN under the same names [174]. In PGM, *distribute-three* is referred to as *consistent-union*, and *union* as *logical-OR* [4]. We refer the reader to these works if such rules are unfamiliar.

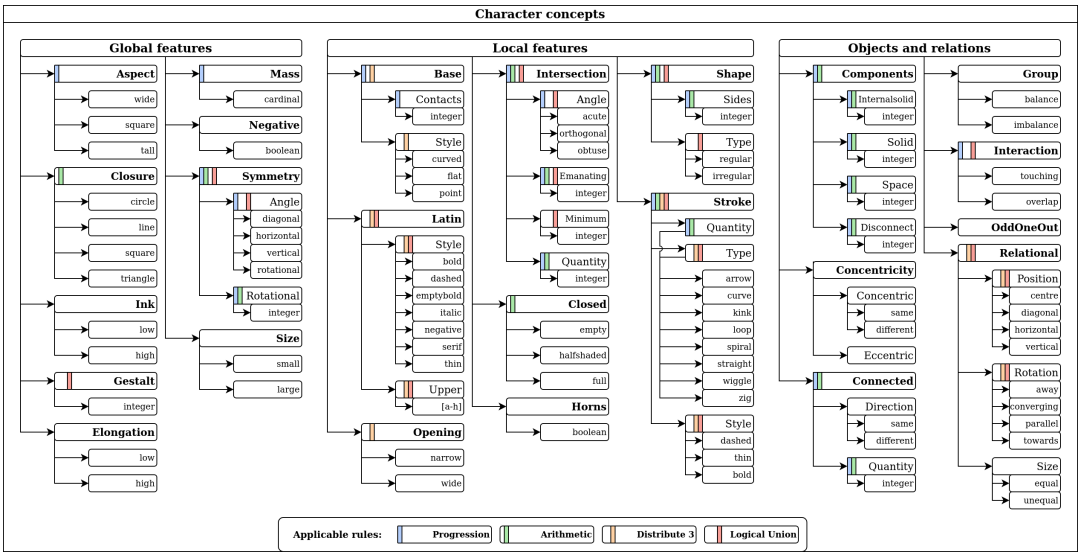


FIGURE 6.3: The conceptual schema of *Unicode Analogies 1.0*, at the time of release. *Constant* problems can be generated using any concept, while the other four rules are applicable to subsets of concepts, as labelled by bars. Concepts appear at multiple levels of hierarchy, beginning with grouping global, local, and relational concepts, becoming finer-grained and specifying the values applicable to each.

6.5.2 Expert annotation

Upon establishing a schema, 4000 characters were pre-selected from sections of Unicode that feature largely symbolic characters.[1] Qualitative suitability criteria include the simplicity and abstractness of characters as an estimate of their amenability to PMPs. Pragmatic criteria include the availability and copyright of fonts to render chosen characters. Manual annotation was then performed by stepping through concepts in the schema and selecting character images for which these concepts were readily perceived.[2] All selections and annotations were made by the first author. This resulted in a final set of over 2500 annotated characters, each possessing 2.8 annotated features on average, with the most polysemic character featuring 20 annotations.

6.5.3 Problem structure and generation

To more directly establish the task as analogy-making, while making efficient use of human annotations, the structure of PMPs in UA differ slightly to those found in RAVEN and PGM, consisting of two rows of context, for a total of nine frames per problem (five context, four answers). This resembles the Visual Analogy format in [65]. PMPs do not exhibit multiple rules, instead, each follows a single rule-concept pair, as the goal is to encourage solvers to use context at the perceptual level. Each frame consists of a single Unicode character rendered at 80x80 binary pixels, which

[1] The full list of Unicode sections is mentioned as supplementary in Section 6.10.

[2] Automated feature extraction algorithms were initially considered. For completeness, we briefly discuss this approach in Section 6.10.2.

is a resolution common to most PMP solvers. Figure 6.4 contains three example PMPs from the dataset to further communicate the problem space.

Generating a new dataset split involves random sampling of the problem space. For each problem requested, a tuple is sampled with the structure *rule-concept-shift* (e.g. *constant-shapesides-noshift*). *Context shift* refers to whether or not both context rows will present the same concept values. For example, a problem that instantiates the *progression* rule over the quantity of dots may do so as two rows depicting 'three, two, one' dots. Requesting context shift would mean the second row altering the progression, e.g. 'one, three, five'. Shifted problems are expected to be more difficult as there are less context frames to evidence the rule.

Upon sampling each problem tuple, context frames are selected to instantiate that tuple in two rows. The last context frame is popped from the list and added to the answers, alongside three foils, all annotated as belonging to the parent concept in the problem tuple. Foils do not depict the concept in a way that would complete the intended rule and invalidate the problem. Additionally, foils cannot be drawn from the problem context, nor can they complete an emergent rule (that is, an unintended but valid alternative rule) as far as the system can infer from character annotations. Finally, the pool of potential foils for any given problem is balanced by only accepting a maximum of three instances of each candidate concept, to ensure that over-represented values (i.e. concepts that apply to relatively large numbers of annotated characters) do not dominate answer distributions and introduce an exploitable bias.

By selecting answers and foils depicting the parent concept, diverse problems can be assembled from orders of magnitude less annotation than naive strategies (i.e. where each image is checked and annotated for each and every feature present in the schema). Answers are also challenging, because the problem rule is guaranteed to be applicable to all candidates, with only one instantiating its concept correctly. Due to the nature of manual annotation, PMPs cannot be guaranteed to always be a) valid (solvable by precisely one candidate answer) or b) human-intuitive in their assembly. Nonetheless, we experimentally confirm that the dataset splits produced by this process remain largely human solvable whilst maintaining a significant challenge for machine solvers, and thereby motivate UA as having utility in exploring this performance gap.

6.5.4 Parameters for defining splits

A the level of defining a single problem, a tuple specifies the rule, concept, and context shift to be instantiated in the problem frames. At the level of defining a dataset split, there are additional parameters that invite experimentation:

- **Rule sampling**. Defines the subset of rules to be made available when sampling problems.

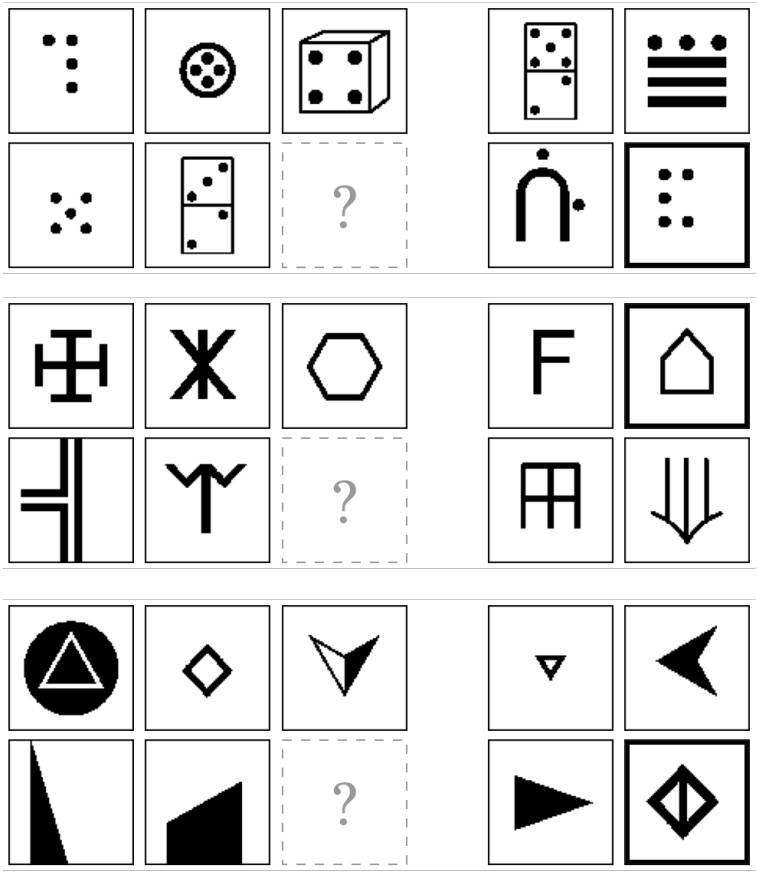


FIGURE 6.4: Three example PMPs from the dataset. The top problem demonstrates a constant number of dots in each row. The middle problem demonstrates arithmetic over the meeting points of lines. The lower problem demonstrates a union; the first two frames depict 3-sided and 4-sided shapes, while the final frames depict both. Correct answers are emboldened.

- **Tuple extrapolation.** Defines which problem tuples to hold out in testing. ***Neutral*** samples all available *rule-class* tuples in both train and test sets. ***Extrapolation*** ensures exclusive tuples across sets. ***Extrapolation-plus*** holds out entire concepts.

- **Context shift.** If true, the generated split will include both shifted and non-shifted problems.

- **Character holdout.** Defines how the character set is split into train and test sets. All problems are generated from these respective sets. If ***None***, all characters are equally available for constructing training and testing problems, but *rule-class-value* tuples are disjoint across splits, to avoid exposing the model to test problems during training. ***Set difference*** holdout ensures train and test character sets are disjoint, requiring models to extract robust features if they are to perform well.

- **Sampling diversity.** ***Balanced*** samples problems uniformly, allowing for equal representation. ***Diverse*** disallows re-instantiating a problem tuple with the same answer to maximise problem diversity and minimise opportunities for memorisation. All experiments use diverse sampling.

6.6 Experiments

We ran five key experiments – *Rule*, *Schema*, *Extrapolation*, *Challenge*, and *Hold-out* – designed to deepen an understanding and appreciation of models' visual conceptualisation abilities. Here, we detail the architectures, dataset splits, and evaluation methods used to facilitate this goal. We also describe the acquisition of a human baseline.

6.6.1 Architectures

We selected three high-performing architectures from the RAVEN literature to fit to each of the dataset splits: the Multi-scale Relation Network [7], the Scattering Compositional Learner [169], and Rel-Base. There exists solvers possessing further structural knowledge for objects and rules [172, 116], obtaining more logical, transparent, and generalisable reasoning at the cost of being bound by strong prior knowledge when asked to correctly parse scenes possessing diverse, overlapping, compound, or gestalt features. We limited experimentation to solvers without such inductive biases, but note that architectures such as the Transformer-based STSN [116] (contemporaneous to this publication) should be investigated further.

To enable modelling the new two-row PMP format by solvers built for three rows, dataset loaders pad problems with empty frames. We noticed no difference in accuracy when comparing this strategy to adjusting architecture input layers. We made use of two baselines as implemented earlier in this thesis: ResNet, and its context-blind variant, used to check for exploitable biases in answer set generation by only viewing answer frames. Instead of treating the blind model as merely a sanity check during development, we subject it to the same tests as all other models to be aware of bias across different splits, as we have done so in previous chapters.

6.6.2 Method and dataset splits used

1. *Rule.* Model versus human performance is explored across all five rule types offered by the dataset. Models are trained and tested on individual rules, as well as jointly trained on all rules, providing average performance. Parameters for defining these splits are set to defaults: Extrapolation is *neutral*, context shift is disabled, and character holdout is *set difference*.

2. *Schema.* Performance is then explored across the three schema subcategories – *Global*, *Local*, and *Objects & Relations* (Figure 6.3) – to provide further understanding of which problem themes were more or less challenging. Parameters for these splits are set to defaults.

3. *Extrapolation.* Models are tested against four extrapolation splits starting with *No Shift* (context-shift disabled), and increasing in difficulty. *Neutral* enables context-shift, while *Extrapolation* and *Extrapolation-plus* also alter the tuple extrapolation parameter to provide finer-grained generalisation results with which to judge the limits of models' extrapolative abilities.

4. *Challenge.* Both easy and challenging concepts are summarised based on a comparison between human and model performance, and the resulting experimentally-informed challenge split is used to further probe the disparity between human and model performance. Parameters are set to defaults.

5. *Hold-out.* The influence of both character hold-out strategies is briefly examined using two splits based on *Constant* rules, in order to exacerbate the effects of non-robust feature learning.

Both model parameter initialisation and dataset seeds needed to be accounted for; the former affects traversal of the optimisation landscape, while the latter affects the distribution of problems across train-test sets, and may prohibit entire problem types from forming across sets in the process of randomly holding out images. To achieve a more robust understanding of model ability, we performed 5-fold cross-validation. Crucially, because dataset splits from this framework aren't amenable to being shuffled and repartitioned (without violating character and tuple holdout), for each dataset split, we generated each fold with random seeds, and trained three randomly-initialised models on each. The size of dataset folds is similar to RAVEN [174], containing 8,000 - 10,000 problems each. All models were trained to a maximum number of epochs given their architecture type, found by preliminary fitting of each model on the *Average* rule set. All materials, including splits, their folds, seeds and other parameters, are made available on our project page.

6.6.3 Establishing a human baseline

In keeping with RAVEN and Bongard-LOGO, we established a baseline of human performance over a set of representative problems to better direct model development. We employed 30 subjects using the Prolific.co research platform, who were remunerated at a rate consistent with the minimum wage in our country. The two selection criteria required subjects to hold a graduate degree, and to form a gender-balanced sample. Subjects were presented with short instructions as to the structure of problems they would encounter, and familiarised with an initial set of presolved problems, sampled from the train set. They were then instructed to complete a set of 15 problems, sampled randomly from the test set. To obtain these sets, we first

generated one fold with a 50-50 train-test split. Across human subjects, at least one problem per potential *rule-concept* tuple was shown. Our experiment was designed using PsychoPy [122] and hosted on Pavlovia.org.

6.6.4 Experimentally informing a new challenge split

By sorting the list of problem types in the fold used for the human baseline, in order of performance difference (human accuracy minus model accuracy), and retaining the top 50%, we establish a *Challenge* split to help guide the development of perception models towards human-like analogical reasoning. In doing so, there is diagnostic potential to uncover weak spots in vision systems and indicate which inductive biases might be necessary to engineer. This also increases the quality of problems, assuming that human accuracy is representative of problem intuitiveness.

6.7 Performance analysis

In this section, we present and analyse the outcomes of our five key experiments. Results reported as accuracy (%).

Rule (Table 6.1). Across the different rule sets, we notice that model accuracy is almost universally below 35%, while humans are still above the top solver on the *Average* (joint) set by 24.4%. This difference is increased to 36.5% over the *Progression* set, which is hypothesised to be due to humans excelling at counting objects (a weakness of deep neural networks [55]). This hypothesis is evidenced by the results of the following experiments, performing more fine-grained analysis on concept types. *Union* problems appear unintuitive for humans, but we believe that this performance could be improved with other experimental designs as the different possible rules were not comprehensively described to participants. While this wouldn't invalidate this data (participants would still be required to perform fluid perception over unseen concepts and characters), it was obtained to serve as a baseline, not a goal to beat. The context-blind solver never achieves more than 5% above what would be expected of random chance, with more advanced architectures only performing within 10% of it, suggesting that overall this dataset succeeds in presenting a significant challenge, while our answer set sampling strategy mitigates exploitable bias.

Schema (Table 6.2). Of the three schema subcategories, the most accurately modelled was *Global*, likely due to concepts such as *ink amount* and *global size* being less abstract and more amenable to feature extraction. Meanwhile, *Object and Relations* saw the human baseline double the leading model's accuracy. Such problems seem to be much harder for machine solvers due to their concepts being abstract and able to be instantiated on a large variety of object types. Unlike many *Global* concepts, which may be partially solvable by pixel counting (e.g. pixels near image borders

Method	Avg	Const	Prog	Arith	Dist3	Union
Blind	27.0	29.5	**29.6**	24.3	28.1	29.7
ResNet	27.4	30.9	26.7	25.7	31.9	30.0
MRNet	**31.1**	33.9	26.8	27.4	34.4	32.9
SCL	28.9	30.1	25.2	25.8	30.7	31.2
RelBase	30.8	**34.5**	28.5	**29.7**	**36.9**	**34.2**
Human	55.5	55.0	65.0	54.0	55.0	42.0

TABLE 6.1: Human vs. model performance across rule types. Average (avg) performance is over the combined test set and is therefore weighted to rule types that have more available concepts.

Method	Global	Local	Obj. & Rel.
Blind	34.0	25.8	25.3
ResNet	35.1	26.5	25.6
MRNet	**39.3**	**30.1**	24.9
SCL	34.1	26.0	24.9
RelBase	39.0	30.0	**26.3**
Human	52.6	58.3	52.2

TABLE 6.2: Human vs. model performance across schema categories.

Method	No Shift	Neutral	Extra	Extra +
Blind	27.0	26.9	26.7	25.6
ResNet	27.4	27.0	27.0	24.9
MRNet	**31.1**	30.2	**28.9**	27.9
SCL	28.9	27.9	27.5	25.7
RelBase	30.8	**31.0**	28.1	**29.5**

TABLE 6.3: Extrapolation performance on datasets with all rules.

might be correlated with *global size*), it is unclear how a network might acquire the features to robustly perceive this.

Extrapolation (Table 6.3). Moving from *No Shift* to *Extrapolation-plus*, we observe a general trend of performance loss across all solvers as expected. With stronger future models, we expect this discrepancy to become even more apparent, as these datasets progressively prohibit memorisation and require solvers to extrapolate learned concepts to increasingly OOD problems.

Challenge (Table 6.4). To our knowledge, the *Challenge* split presents the highest discrepancy between human and machine performance of any PMP dataset in this area, with the leading model trailing 40.2% behind, and most models displaying near-random performance. As clued in by the *Schema* experiment, we continue to

notice that the most successfully modelled concepts belong to the *Global* category. Humans perform very well in counting local features, while models were largely unable to do so.

To further explore how concepts were perceived in problems, Table 6.5 presents the set unions of concepts deemed relatively 'easy' and 'hard', for both human and machine solvers. To obtain these, we first ordered all concepts in the schema by the average accuracy of problems in which they are featured, and then retained the concepts outside the interquartile range. From this, we can see that *Global* problems are often easier for all solvers, while perceiving empty spaces as objects is unintuitive. Not surprisingly, humans perform very well on problems that explore both global and relational object size, whereas models only succeed at global size, further suggesting that their comprehension of size as an abstract concept is limited.

Hold-out (Table 6.6). Comparing models on both *Constant* rule splits – one with character hold-out, and one without – we notice a significant performance increase in some models when the same character set is used to assemble both train and test problems, despite *rule-class-value* tuples being disjoint across splits. This strongly suggests that the use of disjoint character sets is an important design consideration for this framework, and had datasets been constructed without this, we might have critically overestimated model abilities.

A consideration worth mentioning for reproducibility is that models were prone to overfitting, which was partially alleviated by enabling dropout. Given our compute resources, we prioritised k-fold cross-validation with maximum epochs to give a useful first pass of contemporary PMP architectures on this dataset. With hyperparameter tweaking and more nuanced regularisation, along with training schemes such as early stopping and best model selection, additional performance might be achieved. We leave tailoring and developing models to future work. Across all experiments, we notice that despite architectural differences between tested models, similar results were achieved, with the exception of experimentation on different hold-out sets. We believe this observation implies that across models, the same kinds of non-robust features are being extracted, and further motivates the UA challenge by inviting a new class of solvers.

6.8 Broader impact and future work

While this framework is intended to be of primary use for supervised learning techniques in abstract visual reasoning, it is easily extended to new concepts and annotations, inviting future work in artificial intelligence and cognitive science. Investigating the impact of controlling features such as domain shift, distractors and misleading factors, is likely to be of interest in testing models of human concept

Human	Model (RelBase)
Challenge split performance (accuracy and difference)	
71.9%	**31.7%** (-40.2)
Top-5 concepts	
`negative`	`global-size`
`horns`	`negative`
`arrow-quantity`	`ink`
`dash-quantity`	`latin-style`
`internalsolid`	`dash-quantity`
Bottom-5 concepts	
`oddoneout`	`u-quantity`
`opening`	`zig-quantity`
`base-contacts`	`arrow-quantity`
`space`	`interaction`
`uniquesolid`	`uniquesolid`

Challenge split performance, other models	Blind	ResNet	MRNet	SCL
Accuracy	24.8	27.2	28.1	27.7
Difference	-47.1	-44.7	-43.8	-44.2

TABLE 6.4: Breakdown of performance on the *Challenge* split, including a summary of the top and bottom concepts that experimentally informed this split, for human participants and models.

Concepts	Model (RelBase)	
Human	>Q3, 'easy'	<Q1, 'hard'
>Q3, 'easy'	`latin-style`, `negative`, `global-size`, `horns`, `dash-quantity`	`arrow-quantity`, `relational-size`
<Q1, 'hard'	`opening`, `closure`	`space`, `interaction`, `uniquesolid`

TABLE 6.5: Two-by-two table depicting set unions of concepts. These concepts feature in problem types outside the interquartile ranges of human and model performance.

Constant split performance, all models					
H-O	Blind	ResNet	MRNet	SCL	RelBase
None	25.8	31.3	38.5	41.0	**52.2**
Diff.	29.5	30.9	33.9	30.1	34.5

TABLE 6.6: Performance on two *Constant* rule splits, generated with character hold-out set to *None* and *Set difference* (Diff.).

discovery and category learning. There is also the option to run more targeted generalisation experiments: one could test for model numeracy by generating a split with all numeric concepts, but train and test exclusively on arithmetic and progression problems, respectively. Alternatively, one might want to train on local feature concepts, and test for extrapolation to global features. Or, one might implement and test schemas of their own. The released code performs all experiments automatically, given a user-defined schema.

Since we have chosen the concepts used in problem formation, and possess algorithms that are capable of extracting many of these concepts, there is potential to perform more direct probing of concept acquisition in trained models, using methods such as those introduced by [110]. To our knowledge, this has not yet been performed in this area.

Finally, this framework can be adapted for use across different learning paradigms, including meta and unsupervised learning. For example, the Omniglot dataset [93] presents a challenge for few-shot methods aiming to cluster handwritten characters. The problems in Unicode Analogies are generated from an underlying set of annotated polysemic characters, which might pose its own challenge to such methods.

6.9 Conclusion

Of the abstraction datasets that aren't focused on gestalt perception (including all based on Raven's Progressive Matrices), the implication for solvers is that there is a singularly correct way to parse a scene. We argue that testing for analogical reasoning needs to incorporate fine-grained and concept-based analysis, over datasets built to expose non-robust feature learning. We introduce the *Unicode Analogies* challenge, which assembles novel PMPs from diverse and disjoint sets of character images, and brings fluid perception to the progressive matrix format. In doing so, we demonstrate that state-of-the-art solvers are still far from achieving the founding goals their datasets were created for, and encourage new solvers to overcome these limitations. We are excited to see how this framework is adopted by our research community.

6.10 Supplementary

6.10.1 Unicode blocks and fonts used

Our software generates PMPs from an annotated pool of Unicode characters. To select this pool, we looked at blocks/sections in Unicode that contained characters we expected to be amenable to the assembly of novel problems. That is, blocks depicting characters that are symbolic, geometric, not overly complex, and able to express multiple concepts from our schema. We include the full lists of blocks and the fonts used to render them in Tables 6.7 and 6.8. We also include licensing details and websites for all fonts in Table 6.9.

6.10.2 Algorithms for automated feature extraction

While less popular since the advent of convolutional neural networks, there exists much classical work on specialised algorithms for extracting features from thresholded/binary images, with techniques such as skeleton and contour processing (mentioned briefly in Section 2.4.1) being the backbone of many legacy systems. In preliminary work for this chapter, we constructed our own automated feature extraction pipeline to process Unicode characters, with the intention of feeding these features directly to the problem assembly module. Given the breadth of concepts drafted for our schema, and the importance of feature values that more closely aligned with human perception, we ultimately decided to employ manual annotation. For reference, we include a visual summary of the output of our pipeline on two characters, in Figure 6.5. The features we coded for include a percentage of "ink", filled and empty closed contours, closed hulls, measures of centrality, density, ratio and mass offset, as well as symmetry, spaces, internal and total component counts, and local and relative component positions.

6.10.3 Defining custom schemas

Unicode Analogies is intended to be a framework that facilitates the use of PMPs in research across multiple fields. As such, there is the opportunity for researchers to define their own schemas and generate new problems. Encoding a designed schema is straight-forward, and involves typing out a `classes` dictionary consisting of concept keys and applicable rule values, in Python script. The software loads this dictionary and, given parameters to define the split (e.g. sampling diversity), will begin generating problems by loading images directly from their annotation folders on disk. This also makes the annotation process simple, as it requires no additional software or metadata; images need to either be generated and saved to individual folders (or placed there during manual annotation, as was the case with the Unicode characters used in this chapter), and the software will load them and begin assembly.

6.10.4 Further details on training

To promote reproducibility, all data and code, along with scripts to automate all experiments, is released on our project repository. In addition to this resource, we wish to mention further details regarding the training process. MRNet was trained with both cross-entropy and multi-head losses. Blind, ResNet, RelBase, and MRNet models were all trained with dropout on both spatial and fully-connected layers, set to 0.1 and 0.5 respectively. For pragmatic reasons, SCL models were trained using a higher batch size, permitted by relatively fewer trainable parameters. We limited training MRNet to 20 epochs due to overfitting; we expect this is due to it possessing an order of magnitude more trainable parameters than other models. In Table 6.10, we place further information. The average times per epoch are reported on a system using PyTorch's `DataParallel` feature to train models across twin NVIDIA 2080Ti GPUs.

All dataset splits, with the exception of the split used for establishing a human baseline (and subsequent *Challenge* split), possess a train-validation-test ratio of 70-3-27. This can be easily changed in software to facilitate different experiments. Note that this split ratio simultaneously defines both the ratio of characters used to form problems in each partition, as well as the ratio of problems formed. In *Extrapolation / Extrapolation-plus* splits, it also defines the ratio of held-out problem tuples / class types.

The number of problems available to each split is highly dependent on the split parameters used. If there are weak restrictions governing character hold-out and sampling diversity (for instance, if no hold-out is requested, or if the system is permitted to re-instantiate a problem tuple with the same answer), then the full number of problems requested (default = 10000) will likely be generated. Given that character hold-out and sampling diversity are key design considerations and both greatly contribute to this dataset's challenging nature, some splits will not make up the full 10000 problems. While most splits remained between 8000-10000 as reported earlier, the *Union* split is only able to generate on average, 2300 problems per fold. This is due to answers to union problems being more scarce (they need to depict the right intersection of concepts), as well as there being a limited number of concepts in the schema applicable to union problems. The training partitions of all dataset splits contain both context-shifted and non-shifted problems, regardless of whether validation and testing is requested to contain shifted problems, as this enables the generation of far more problems (and assumedly, training more capable models) given limited annotations.

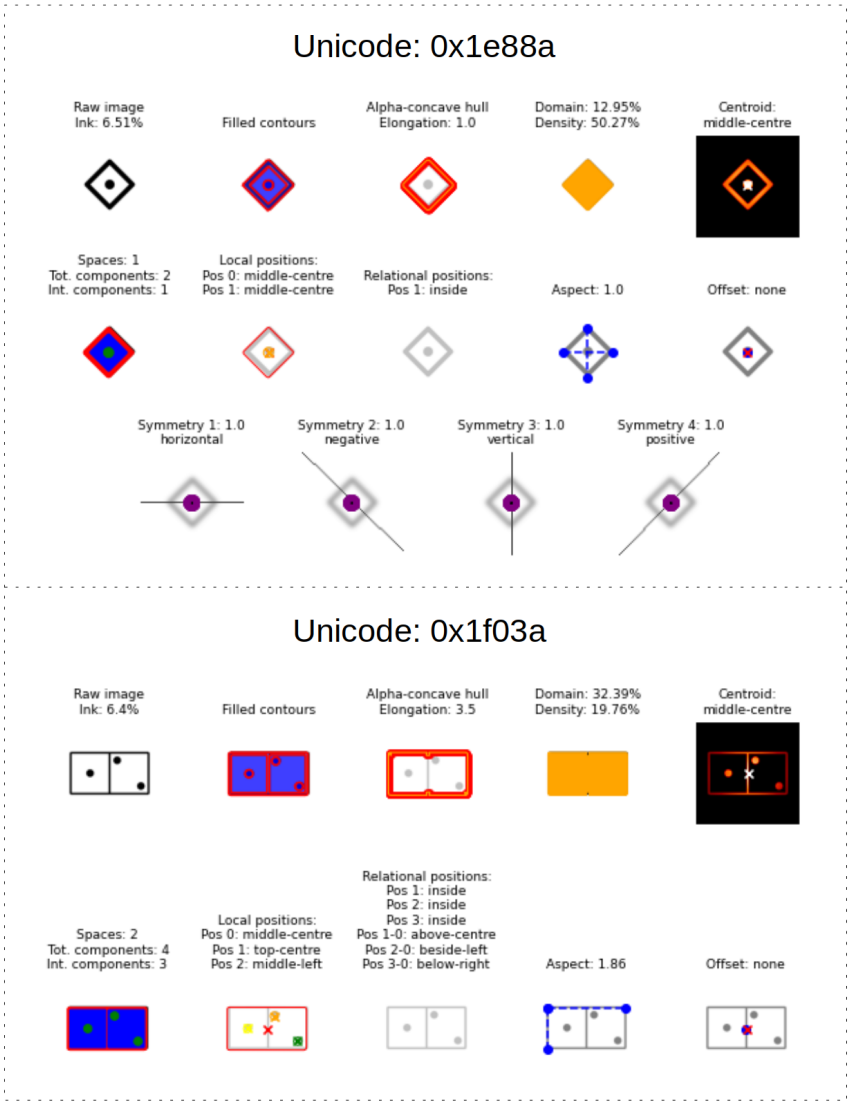


FIGURE 6.5: The result of Unicode characters 0x1e88a (top) and 0x1f03a (bottom) being processed by an automatic feature extraction pipeline.

Unicode block	Name	Rendering font
16A0-16FF	Runic	Alphabetum
10C80-10CFF	Old Hungarian	"
11000-1107F	Brahmi	"
10100-1013F	Aegean Numbers	"
10300-1032F	Old Italic	"
103A0-103DF	Old Persian	"
10800-1083F	Cypriot Syllabary	"
12400-1247F	Cuneiform Numbers and Punctuation	"
A6A0-A6FF	Bamum	Google Noto Bamum
16800-16A3F	Bamum Supplement	"
A000-A48F	Yi Syllables	Google Noto Yi
A490-A4CF	Yi Radicals	"
2D30-2D7F	Tifinagh	Google Noto Tifinagh
10600-1077F	Linear A	CTAN Linear A
1E800-1E8DF	Mende Kikakui	Mende Kikakui
1BC00-1BC9F	Duployan	Duployan
16F00-16F9F	Miao	Miao Unicode
1D800-1DAAF	Sutton SignWriting	Sutton SignWriting
10380-1039F	Ugaritic	Quivira
680-169F	Ogham	"
1400-167F	Unified Canadian Aboriginal Syllabics	"
18B0-18FF	Unified Canadian Aboriginal Syllabics Extended	"
A700-A71F	Modifier Tone Letters	"
0000-007F	Basic Latin	Symbola 10.24
0080-00FF	Latin-1 Supplement	"
0100-017F	Latin Extended-A	"
0180-024F	Latin Extended-B	"
0250-02AF	IPA Extensions	"
02B0-02FF	Spacing Modifier Letters	"
0300-036F	Combining Diacritical Marks	"
0370-03FF	Greek and Coptic	"
0400-04FF	Cyrillic	"
0500-052F	Cyrillic Supplement	"
2000-206F	General Punctuation	"
2070-209F	Superscripts and Subscripts	"
20A0-20CF	Currency Symbols	"
20D0-20FF	Combining Diacritical Marks for Symbols	"
2100-214F	Letterlike Symbols	"

TABLE 6.7: Unicode blocks used, with the fonts used to render them.

Unicode block (continued)	Name	Rendering font
2150-218F	Number Forms	Symbola 10.24
2190-21FF	Arrows	"
2200-22FF	Mathematical Operators	"
2300-23FF	Miscellaneous Technical	"
2460-24FF	Enclosed Alphanumerics	"
2500-257F	Box Drawing	"
2580-259F	Block Elements	"
25A0-25FF	Geometric Shapes	"
2600-26FF	Miscellaneous Symbols	"
2700-27BF	Dingbats	"
27F0-27FF	Supplemental Arrows-A	"
1D000-1D0FF	Byzantine Musical Symbols	"
1D100-1D1FF	Musical Symbols	"
1D00-1D7F	Phonetic Extensions	"
2440-245F	Optical Character Recognition	"
27C0-27EF	Miscellaneous Mathematical Symbols-A	"
2800-28FF	Braille Patterns	"
2900-297F	Supplemental Arrows-B	"
2980-29FF	Miscellaneous Mathematical Symbols-B	"
2A00-2AFF	Supplemental Mathematical Operators	"
2B00-2BFF	Miscellaneous Symbols and Arrows	"
2E00-2E7F	Supplemental Punctuation	"
4DC0-4DFF	Yijing Hexagram Symbols	"
FE20-FE2F	Combining Half Marks	"
1D200-1D24F	Ancient Greek Musical Notation	"
1D300-1D35F	Tai Xuan Jing Symbols	"
1D360-1D37F	Counting Rod Numerals	"
1D400-1D7FF	Mathematical Alphanumeric Symbols	"
1F100-1F1FF	Enclosed Alphanumeric Supplement	"
1F300-1F5FF	Miscellaneous Symbols and Pictographs	"
1F700-1F77F	Alchemical Symbols	"
1F780-1F7FF	Geometric Shapes Extended	"
1F030-1F09F	Domino Tiles	"
1D2E0-1D2FF	Mayan Numerals	Babelstone Han
3000-303FCJK	Symbols and Punctuation	"
FE30-FE4FCJK	Compatibility Forms	"
1FB00-1FBFF	Symbols for Legacy Computing	Legacy Computing Font

TABLE 6.8: Unicode blocks used, with the fonts used to render them (continued).

Rendering font	Licence	Website
Alphabetum	Paid publishing licence	http://guindo.pntic.mec.es/ jmag0042/alphaeng.html
Google Noto Bamum	OFL 1.1	https://fonts.google.com/
Google Noto Yi	"	"
Google Noto Tifinagh	"	"
CTAN Linear A	The LATEX Project Public Licence	https://ctan.org/pkg/lineara
Mende Kikakui	OFL 1.1	https://athinkra.github.io/mende-kikakui/
Duployan	"	https://github.com/dscorbett/duployan-font
Miao Unicode	"	https://github.com/phjamr/MiaoUnicode
Sutton SignWriting	"	https://slevinski.github.io/SuttonSignWriting/
Quivira	Public domain / unrestricted	http://www.quivira-font.com/
Symbola 10.24	Freeware ($\leq$ Symbola 10.24)	https://packages.fedoraproject.org/pkgs/gdouros-symbola-fonts
Babelstone Han	OFL 1.1	https://www.babelstone.co.uk/Fonts/Han.html
Legacy Computing Font	"	https://github.com/dokutan/legacy_computing-font/

TABLE 6.9: Fonts used, with their licences and website details.

Model architecture	Learning rate	Batch size	Max. epochs	Avg. time per epoch (s)
Context-blind	3e-4	32	60	2.45
ResNet	3e-4	32	60	3.73
RelBase	3e-4	32	60	9.97
MRNet	1e-3	32	20	19.15
SCL	1e-3	128	60	3.64

TABLE 6.10: Further information on hyperparameters used in training models.

Chapter 7

Conclusion

As this is written, it is the beginning of 2024, and the interested layperson is trying to reconcile their expectations of *reasonable* machines with a market proliferation of LLMs that, to them, seem to make mistakes computers shouldn't make. While vastly more capable — at least, in a general capacity — than any system before them, to use the language of 2D semantics from Chapter 2, we are always one critical intension away from exposing the gulf of meaning between us and our systems. So, we have asked: how can we ensure models acquire concepts that will allow them to generalise as we do? If a system cannot find these concepts, it not only remains unable to traverse distributional shifts, but it will consume far more data to get even a fraction of the way. While this can be of humorous value when conversing with ChatGPT, it will work against de-biasing education and politics, and prohibit usage for critical tasks in medical diagnostics and defence.

Humans find *elegant* concepts, and while this can be a suitcase word, this thesis has used it in the sense that Occam's razor is elegant. We don't multiply entities beyond necessity, we look for simple explanations. This happens not just at abstract levels, like trying to find a scientific theory, but for low-level perception, too. This is an inductive bias that machines demonstrably do not possess to the degree we do. But, there is an important distinction to be made here; what constitutes a simple explanation to a human is rendered so by an elaborate web of meaning. For example, the complexity of an entire field of research is shorthanded by $e = mc^2$. Yet, this equation is an abstraction that has given humanity phenomenal predictive power. Elegance does not imply simplistic thought.

So, on the face of it, concepts acquired by machine systems are not necessarily inelegant in an Occam's razor sense: as we have shown, they are perfectly capable of finding simple shortcuts and exploiting datasets with surface-level features. The key difference is that they carry many, many such concepts, piecing together their worlds in this way, without broad coherence. Our current species of machines are not the Bongardian martians of Section 2.1; instead of attempting to assemble cars from a blueprint of atomic detail, they subsist on countless, varied blueprints to compensate for an inability to properly draw up new ones. As such, we have characterised this as *concept vampirism*. Like the Bayesian model in Chapter 5, humans employ complexity where needed — striking a trade-off for increased explanatory

power — allowing us to form more comprehensive, integrated, and stable pictures of our environments. In a manner of speaking, machine systems take the watercourse way, towards least resistance, accumulating knowledge without a deeper sense of meaning as per Chapter 2. For now! There is active debate in our community as to whether or not the current trajectory of foundation model development will see this shortcoming rectified. This thesis has argued that, regardless of the approach, there is more than ever, a need to understand our systems' reasoning processes at fundamental levels, *piercing layers of imitative sophistry*, such that we can continue giving them increased responsibilities as we cooperatively shape society.

Our work has championed the use of PMPs as a powerful diagnostic tool for abstract reasoning, and has furthered the methodology for their application in machine psychometrics. These problems should be thought of as *visual microworlds for scientific agents*, as they ask the solver to engage multiple kinds of inferential processes towards completing sequences in sensible ways. The PMPs contributed by this thesis have been in service of the lofty goal of distilling the core problem of AI, in the expectation that as our community progresses abstract reasoning, we get closer to achieving general thinking machines.

7.1 Contributions

We discuss the contributions of this thesis by first recalling our research questions once more, as defined in Section 1.2:

Core: *How can we develop vision models to learn the "right" concepts, allowing them to generalise to new scenarios?*

We have deepened our appreciation of the core question by identifying the reasoning processes thought to give rise to general perception, aligning our benchmarks to be a) more discriminative of these processes, and b) less exploitable by solvers that employ non-robust concepts. Not only have we seen our two architectures from Chapter 3 — Rel-Base and Rel-AIR — become increasingly obsolete with the release of Bayesian-RAVEN and Unicode Analogies, we have actively endeavoured in this direction. Both architectures possess inductive biases that are rewarded on RAVEN and I-RAVEN. Rel-Base implements separate encoders for frame-level and sequence-level perception, in order to disentangle their operation. On a dataset such as RAVEN, this is very effective at increasing data efficiency and generalisation performance. Likewise, Rel-AIR employs an object-centric inductive bias, which is met with further performance gains. However, we are pleased that these biases are ultimately of the kind that struggle to progress the UA challenge at all, as it is a sign that this thesis has significantly pushed the state of PMPs to new territory.

Bayesian-RAVEN and UA are based on investigations that identify non-robust feature memorisation strategies. By introducing rule ambiguity to RAVEN, Bayesian-RAVEN asks solvers to contextually reassess rules as scored by a Bayesian oracle, instead of merely classifying them. By introducing perceptual ambiguity, UA asks solvers to find a way to perceive rules in highly diverse stimuli, drastically expanding the hypothesis space, and blurring the delineation between perception and reasoning. In doing so, Bayesian-RAVEN and UA contribute towards reinstating inductive and abductive forms of inference at test time, to a task that had been reduced to deduction (classification) by SOTA solvers that could achieve near-perfect accuracy, but still not generalise in the ways the community had hoped.

UA was created in the desire to merge principles of RPMs and Bongard problems into the one format, in order to have a multiple-choice psychometric task that would also incorporate Bongard's work in capturing a core problem of perception. Bongard realised that gestaltism pointed to a quality of perception that evaded machine classification theories of his time [11]. UA re-establishes Bongard's critique today, with the attitude of directing research efforts towards progressing his problem. The hypothesis is that, in solving this, abilities over real-world applications should follow. While this may indeed be solved as a by-product of those applications, we believe that for efficiency and clarity, it is sensible to strip the problem of distractions as much as possible.

Each chapter contributed towards multiple research questions, in service of the core question above. We now address each question given these contributions.

RQ1: To what extent can deep learning techniques be considered capable of performing abstract reasoning, of the kinds associated with humans?

Chapter 2 reviewed the current theories of conceptual abstraction and analogical reasoning, to more fully appreciate the nature of concepts acquired by humans. We recognised that concepts allow an agent to perceive relatively stable structure across diverse environments, empowering them to make accurate predictions towards goal acquisition. Isolating fundamental differences in the way that humans and deep-learnt systems build up knowledge of the world, we introduced the notion of *concept vampires*, commenting on our propensity to ascribe general reasoning ability to agents that can perform seemingly general tasks. We stated that this biases us to underplay the reality of shortcut learning and dataset exploitation, which can lead to serious consequences. Chapters 3 and 4 picked up this thread by sense-checking the value of PMPs in light of the reality of shortcut learning. Offering a functional construction of the forms and interplay of inferential processes, we assessed the kinds of reasoning they require, and suggested ways to improve their diagnostic value. This set the foundations for the creation of own PMP datasets in the investigations to follow. Chapter 5 executed a number of experiments, testing for induction, generalisation, and brittleness, to further locate differences in reasoning between humans

and deep learning models. Chapter 6 continued to compare humans and machines on a whole new suite of experiments, and with a selection of machine architectures, breaking down performance by fine-grained, concept-based analyses not able to be performed with RAVEN or Bayesian-RAVEN. Our findings support the conclusion that, despite the success of SOTA models on RAVEN, due to brittleness arising from non-robust feature extraction, there is still much work to be done in closing the performance gap between models and humans.

RQ2: What architectures, inductive biases, datasets, and curricula, might advance the acquisition of such abilities in vision systems?

Chapter 2 commented on the controversy of hand-feeding our systems expert knowledge, along with a review of candidate inductive biases. We discussed *elegance* as a bias that is core to the human condition, and grounded this via the Bayesian brain hypothesis. Chapter 3 introduced two SOTA architectures, Rel-Base and Rel-AIR, that are the results of applying inductive biases to improve generalisation performance on RAVEN. Rel-Base is the first in the literature to consistently exceed human-level performance, providing the thesis with a general-purpose solver, while Rel-AIR was the first to apply unsupervised scene decomposition to solving PMPs. Chapter 5 then introduced Bayesian-RAVEN, relaxing the well-structured property of RAVEN problems, asking solvers to integrate further sources of information. We defined a Bayesian oracle to label problems, and showed it to be reasonably predictive of human responses as well. Chapter 6 took this one step further, moving from the rule ambiguity of Bayesian-RAVEN, to perceptual ambiguity, dismantling the simple geometry of RAVEN, and replacing it with a far richer conceptual schema.

RQ3: What methodological changes are required, such that we may measure and evaluate these abilities more comprehensively?

Chapter 2 opened a conversation between AI and semantics, identifying the role of analogical reasoning in the formation and ongoing shaping of concepts, and suggested that shared analogy tasks may be an advisable way forward in the pursuit of common understanding between humans and machines. We therefore promoted PMPs as an ideal test bed, and motivated the methodological areas in need of development in order to successfully import them into our field, such as avoiding the anthropocentrism of classical psychometrics, as well as the trap of chasing accuracy on in-domain test splits. Chapter 3 picked up this thread by investigating RAVEN's suitability as a benchmark to compare human and machine reasoning, noticing there was little work in testing generalisation performance. We uncovered a major bias in RAVEN, meaning that several results in this area became invalid, which prompted a re-evaluation and highlighted the reality of shortcut-learning. Chapter 4 therefore argued that our default posture should be to assume a null hypothesis: that a given model has not found concepts that will generalise OOD until reasonable evidence

is supplied to suggest otherwise. We implemented targeted baselines for hunting shortcuts, and discussed the strategic use of hypothesis competition to avoid problems rewarding naive solution strategies. We confronted the 'elephant in the room': that state-of-the-art solvers do not extrapolate without breaking, and yet otherwise appear to perform better than humans. We therefore argued that the goal of this area should be to build and evaluate systems that can demonstrate the dexterity required to navigate distributional shifts. Towards this goal, Chapter 6 started with the position that objectivism is counter-productive to AI research, since training systems to see things only one way — "this is a white circle", or, "this is a black square" — does not capture the way humans creatively use concepts in an *ad hoc* fashion. The problems in UA are therefore polysemic, based on gestalt principles, propelling PMP research in the direction of Bongard.

7.2 Future directions

Bayesian-RAVEN serves to fulfil a set of experiments not otherwise attempted with the RAVEN set, investigating abstract visual reasoning models through a Bayesian lens. We limited the scope of that work to comparing humans, oracles, and solvers inheriting the same architecture (Rel-Base), in order to establish the dataset. Therefore, there are further experiments to be done, benchmarking other solver architectures in the RAVEN literature. As this is a new direction, it also invites further research in artificial intelligence and cognitive science, continuing to investigate the intersection of Bayesianism and computer-generated PMPs in evolving both human and machine psychometrics.

Meanwhile, Unicode Analogies is an extensible framework that can be adapted to entirely new conceptual schemas. Like Bayesian-RAVEN, this also facilitates future experiments. UA demonstrates that SOTA solvers are still far from achieving the founding goals their datasets were created for, encouraging new solvers to overcome such limitations. Additionally, as foundation models only made their debut during the author's candidature, they are mentioned in Chapter 2 for completeness, but the thesis itself restricts its scope to vision models trained directly on the datasets themselves. All problems in Bayesian-RAVEN are labelled with text representations for future work with LLMs, while cutting-edge multimodal systems such as GPT-4V[1] or Gemini-Ultra[2] might attempt the UA challenge. Good luck!

If there were to be a *Chapter 8*, it would undoubtedly be on applying the principles of a) backing models into corners, b) anti-objectivism, and c) universal induction, for three outcomes. The first is that Bayesian-RAVEN and UA both tackle related problems, but there remains the opportunity to create a dataset with the strengths of both, finding the "goldilocks" problems theorised in Section 5.7 that will strike the perfect trade-off between formedness and discriminative utility. For this,

[1] https://openai.com/research/gpt-4v-system-card
[2] https://blog.google/technology/ai/google-gemini-ai/

our Bayesian measure of confidence can be broadened to estimate problem quality, potentially informing automated problem generation and puzzle design. The second, is that our definition of a prior distribution in Chapter 5 — while acknowledging the value of a more universal direction — is begging to be developed in a more mathematically rigorous way, following the insights of Solomonoff induction [147, 75]. This would allow researchers to make even stronger claims about the suitability of their PMPs to test for general intelligence. The third: UA's caveat is that, while its problems are auto-generated, its conceptual schema is still hand-picked. This means that, unlike Bongard's original problems, there are no guarantees on problem quality. Assembling problems from pre-annotated frames results in PMPs that vary widely with regards to the elegance of concepts expressed.

While pursuing these principles is not trivial, we expect that such a direction will bring us closer to capturing the open-ended nature of Bongard problems, with which we could form the basis of a much more rigorous psychometric practice.

7.3 Closing thoughts

PMPs are deceptively simple, and powerfully discriminative. They can draw out hidden weaknesses of sophisticated models using only abstract line drawings. This thesis set up camp in nascent research territory, making broad contributions to the ongoing work of importing this problem format into deep learning. As AI technologies become a fixture of civilisation, we are hopeful for the future of machine psychometrics, towards the development of systems that may one day perceive the world in more splendid and nuanced ways than their creators.

Appendix A

Sample PMPs from UA

To further aid researchers in exploring the Unicode Analogies dataset and its current conceptual schema, we provide just under a hundred auto-generated PMPs, each with a description of the depicted rule and concept, and the correct answer frame. Answers are numbered from 1 to 4, starting top left and continuing left to right, top to bottom. All PMPs shown here are generated without context shift, to more easily communicate the base rule.

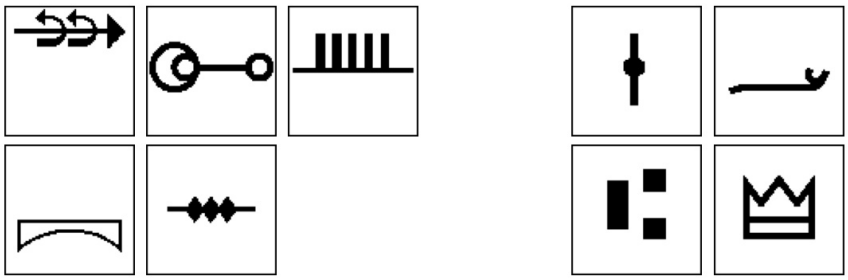


FIGURE A.1: Constant aspect ratio (wide). Answer=2.

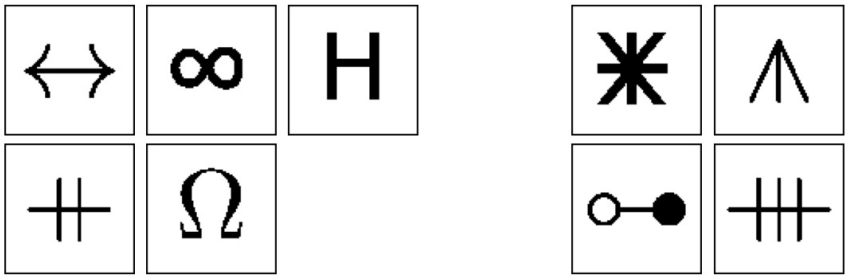


FIGURE A.2: Constant number of base contacts (two). Answer=3.

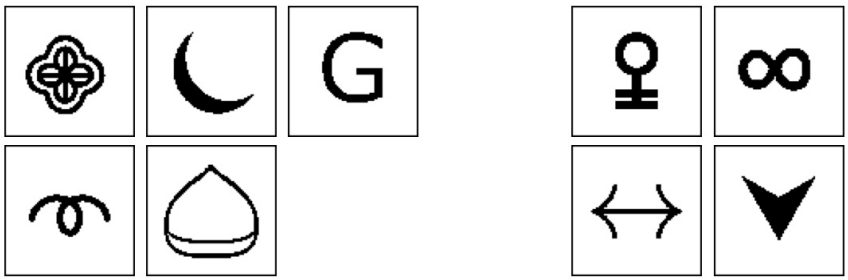


FIGURE A.3: Constant base style (curved base). Answer=2.

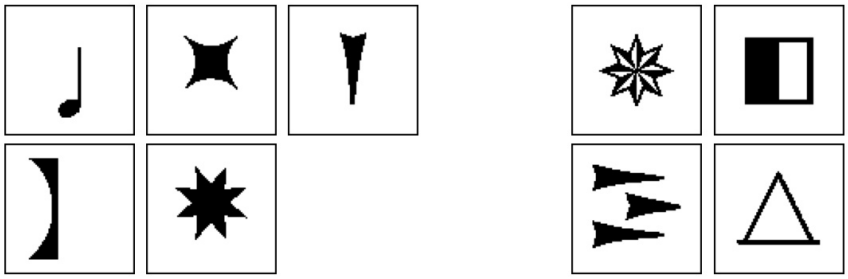


FIGURE A.4: Constant closed fill (full). Answer=3

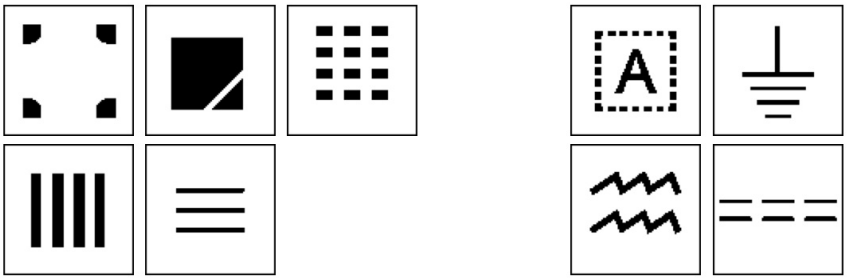


FIGURE A.5: Constant closure pattern (square). Answer=1.

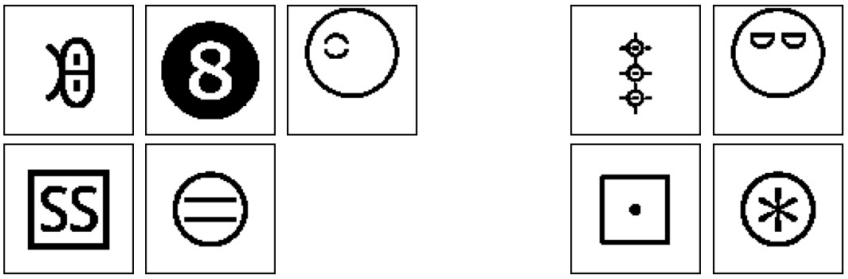


FIGURE A.6: Constant number of internal solid components (two). Answer=2.

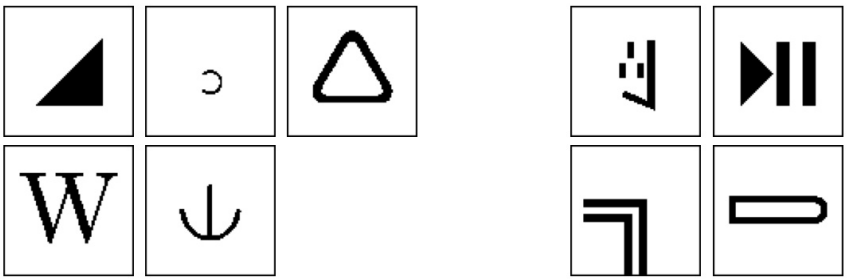


FIGURE A.7: Constant number of total solid components (one). Answer=4.

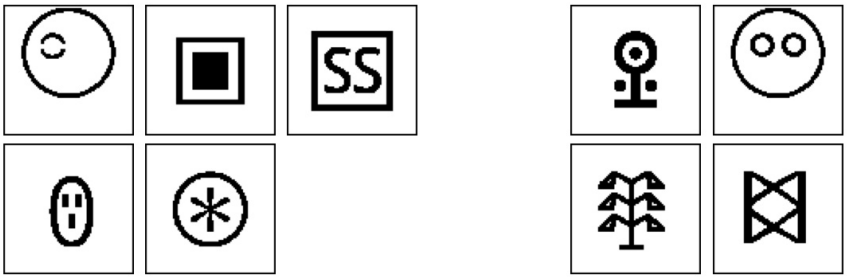


FIGURE A.8: Constant number of internal spaces (one). Answer=1.

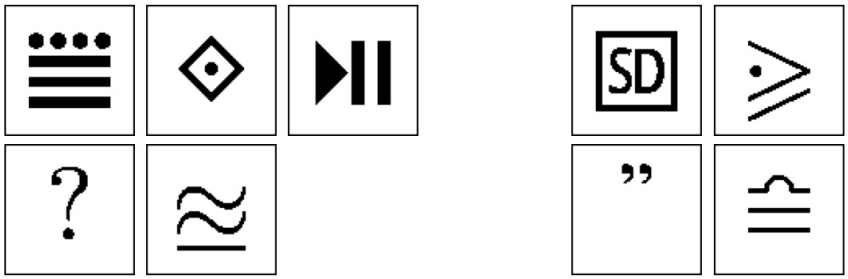


FIGURE A.9: Constant number of unique solid components (two). Answer=4.

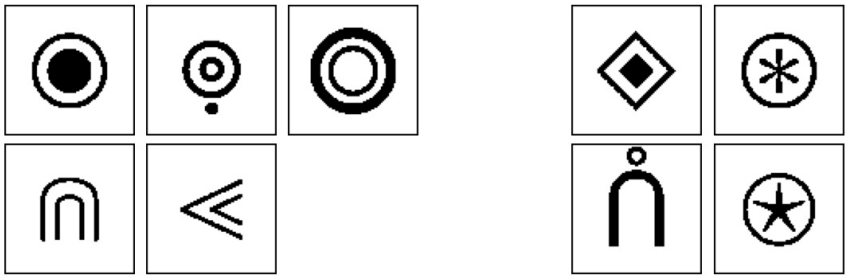


FIGURE A.10: Constant in exhibiting concentric shapes. Answer=1.

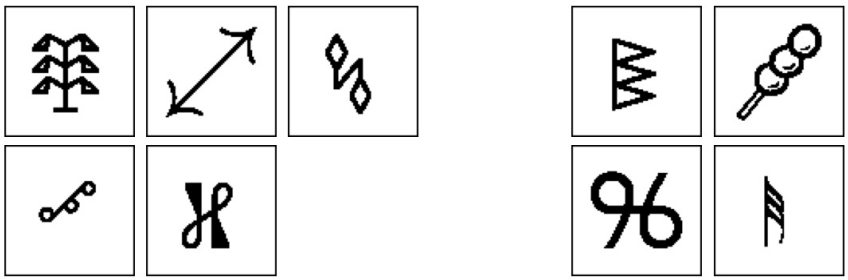


FIGURE A.11: Constant in exhibiting connected items facing different directions. Answer=3.

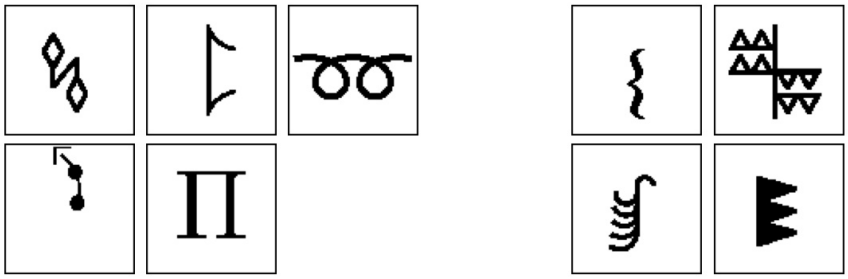


FIGURE A.12: Constant number of connected items (two). Answer=1.

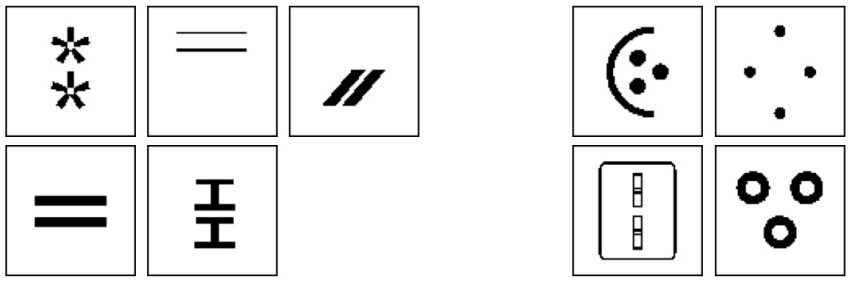


FIGURE A.13: Constant number within groups (two). Answer=3.

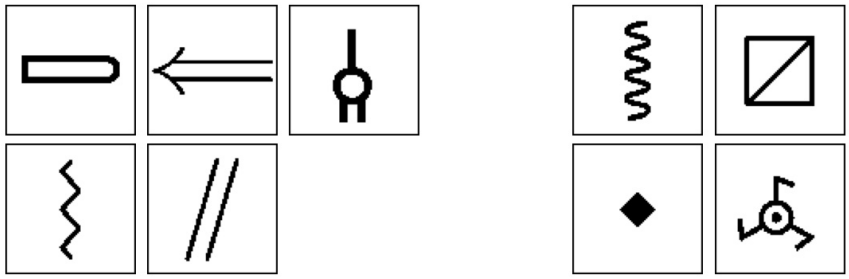


FIGURE A.14: Constant elongation (high). Answer=1.

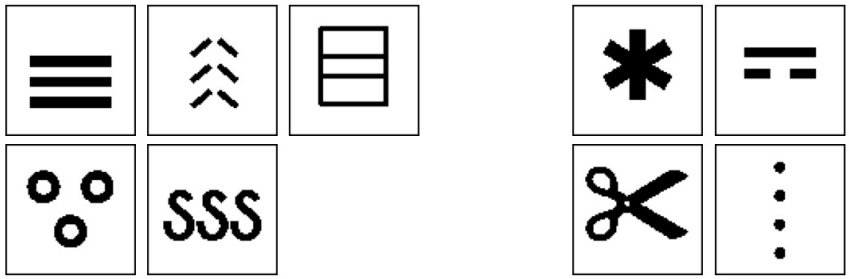


FIGURE A.15: Constant gestalt number (three). Answer=2.

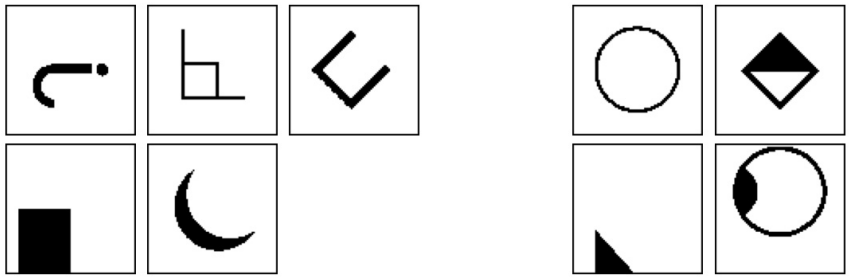


FIGURE A.16: Constant global mass centroid (South-West). Answer=3.

FIGURE A.17: Constant global size (large). Answer=1.

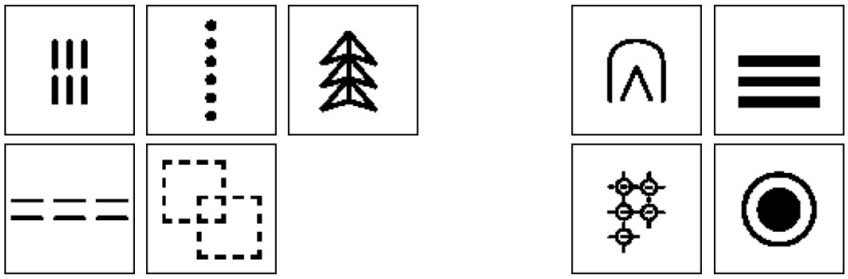


FIGURE A.18: Constant in exhibiting "balanced" groups (same object and with symmetry). Answer=2.

FIGURE A.19: Constant horns. Answer=2.

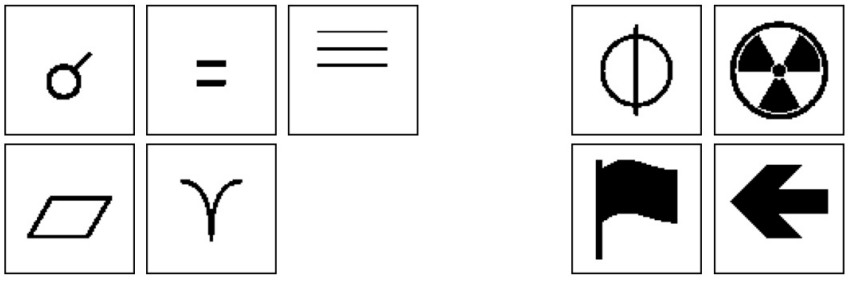


FIGURE A.20: Constant ink level (low). Answer=1.

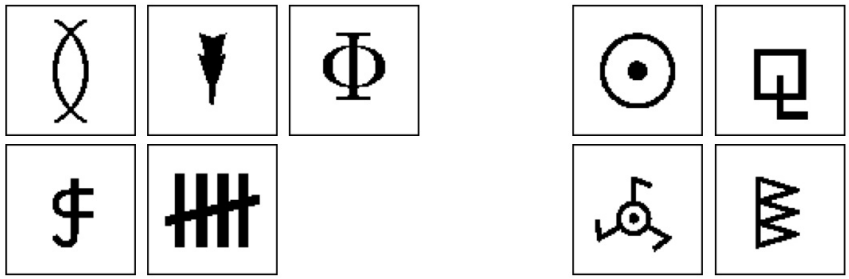


FIGURE A.21: Constant interaction (overlap). Answer=2.

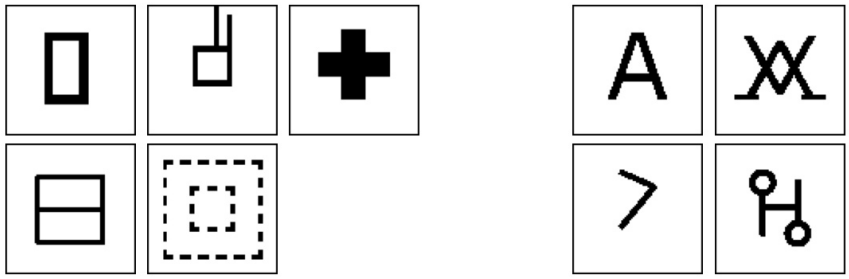


FIGURE A.22: Constant in exhibiting a type of intersection (orthogonal). Answer=4.

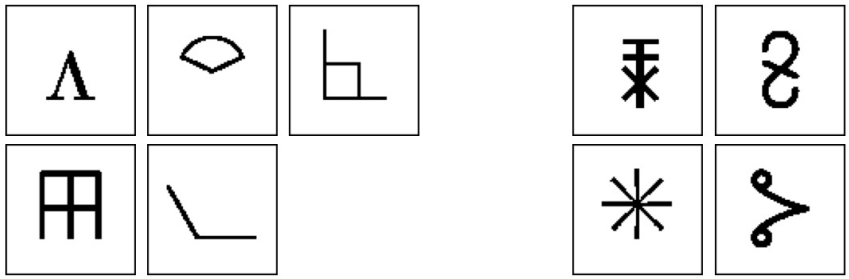


FIGURE A.23: Constant in exhibiting an intersection with a number of emanating lines (two). Answer=4.

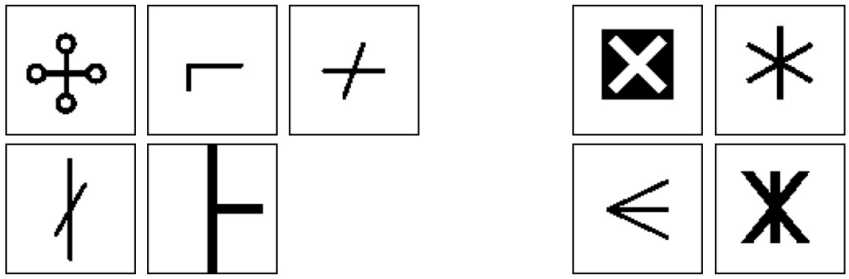


FIGURE A.24: Constant in exhibiting an intersection with a minimum number of lines (two). Answer=1.

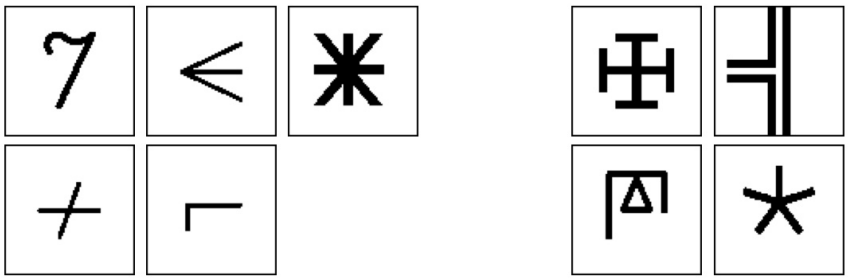


FIGURE A.25: Constant number of intersections (one). Answer=4.

FIGURE A.26: Constant in character style (bold). Answer=3.

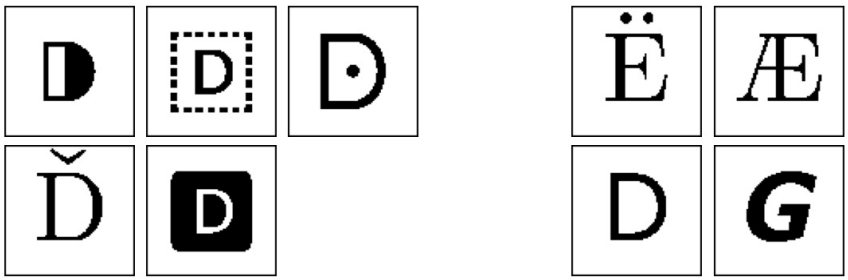


FIGURE A.27: Constant Latin character (uppercase-D). Answer=3.

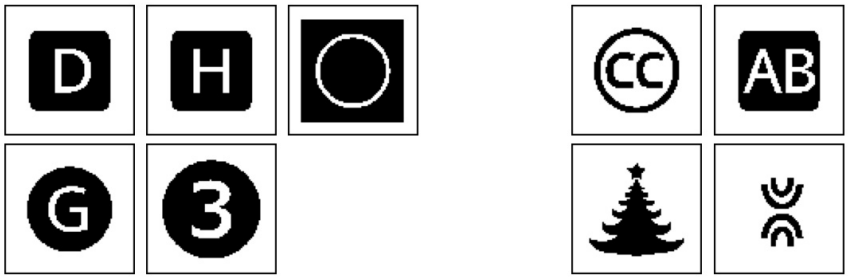


FIGURE A.28: Constant in exhibiting negative space. Answer=2.

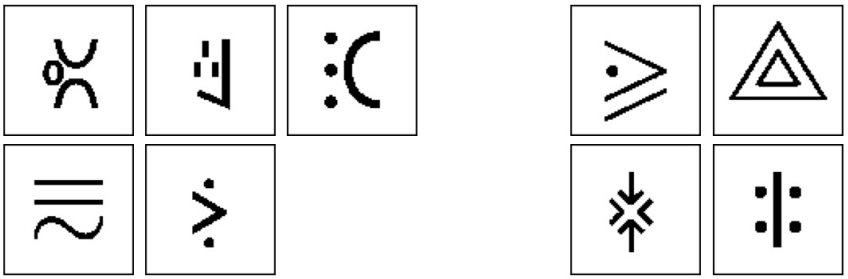


FIGURE A.29: Constant in exhibiting an odd-one-out scenario. Answer=4.

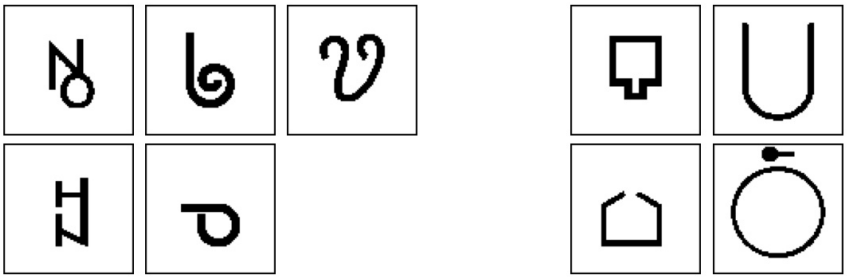


FIGURE A.30: Constant width of opening (narrow). Answer=3.

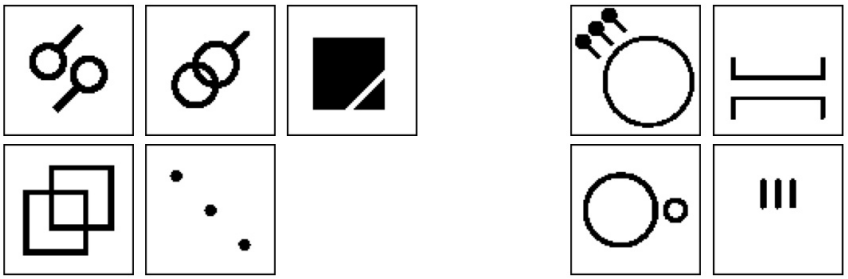


FIGURE A.31: Constant in exhibiting a type of relational position (diagonal). Answer=1.

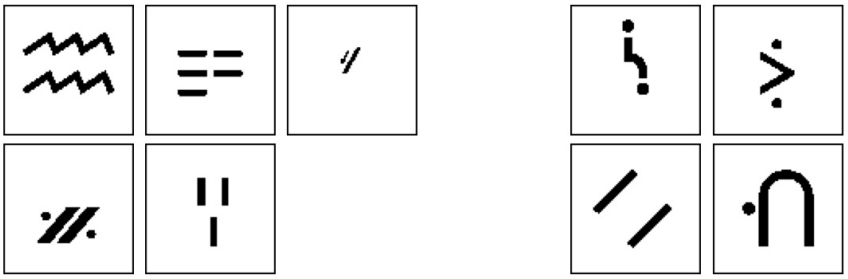


FIGURE A.32: Constant in exhibiting a type of relational rotation (parallel). Answer=3.

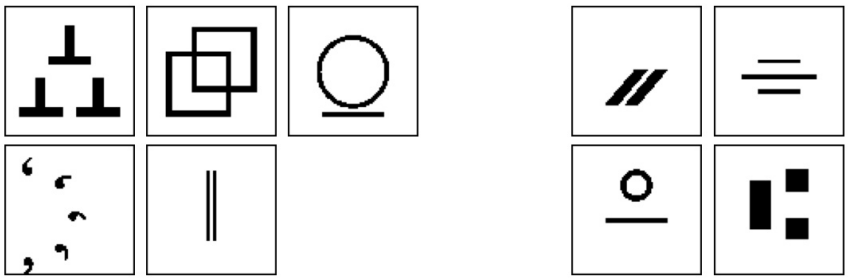


FIGURE A.33: Constant in relational size (equal). Answer=1.

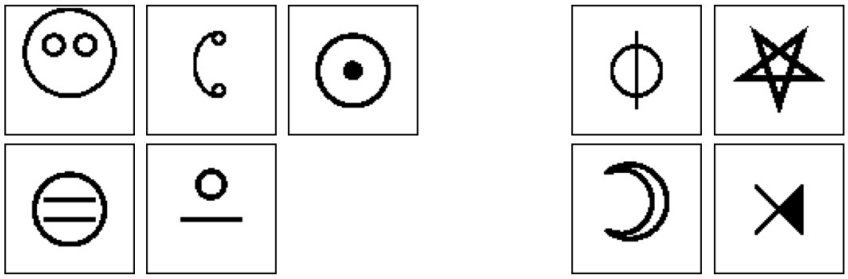


FIGURE A.34: Constant in number of shape sides (one). Answer=1.

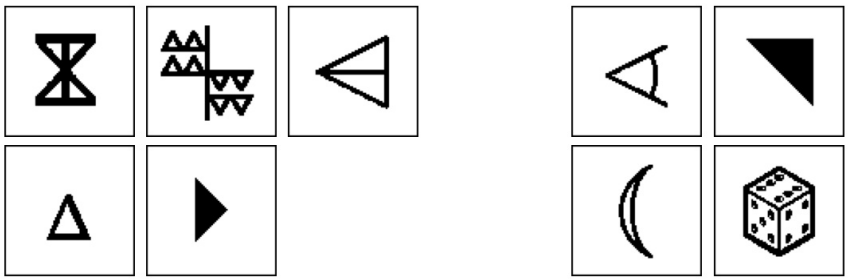


FIGURE A.35: Constant in shape type (triangle). Answer=2.

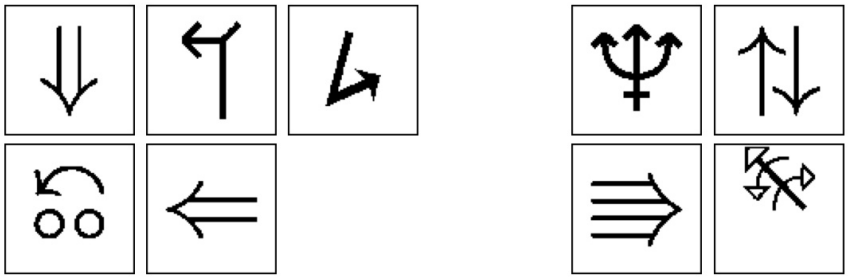


FIGURE A.36: Constant number of arrows (one). Answer=3.

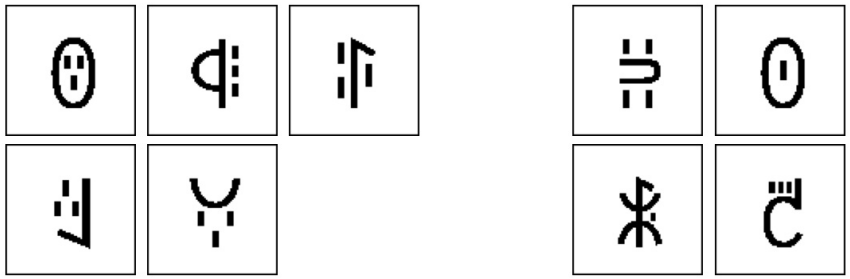


FIGURE A.37: Constant number of dashes (three). Answer=4.

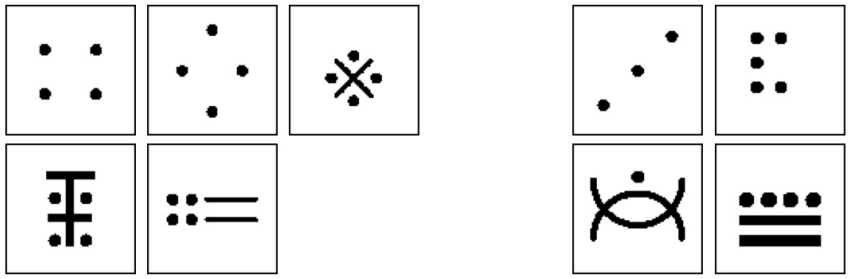


FIGURE A.38: Constant number of dots (four). Answer=4.

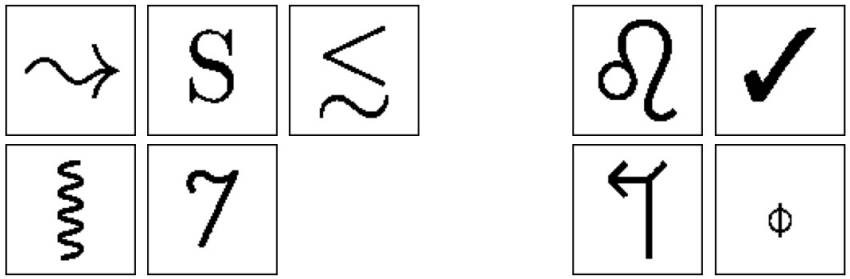


FIGURE A.39: Constant in exhibiting a particular stroke feature ("wiggle"). Answer=1.

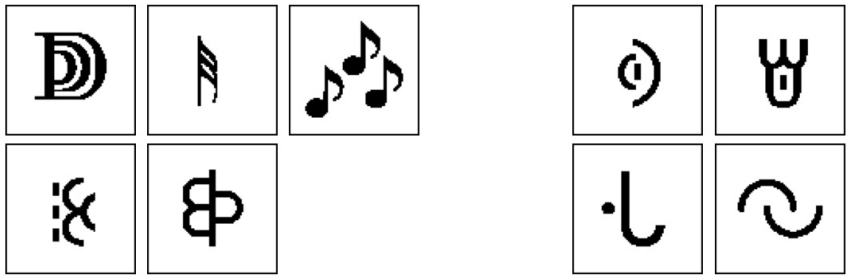


FIGURE A.40: Constant number of curved line strokes (three). Answer=2.

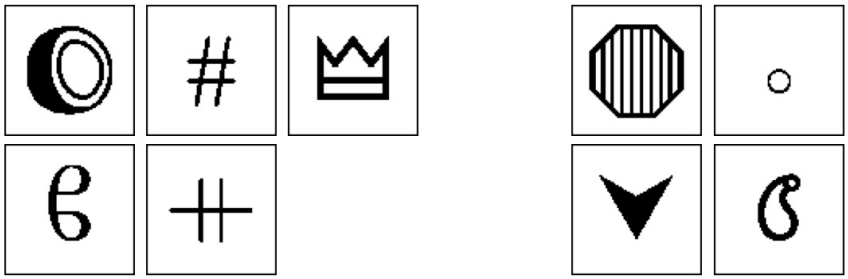


FIGURE A.41: Distribute three base style (curved, point, flat). Answer=1.

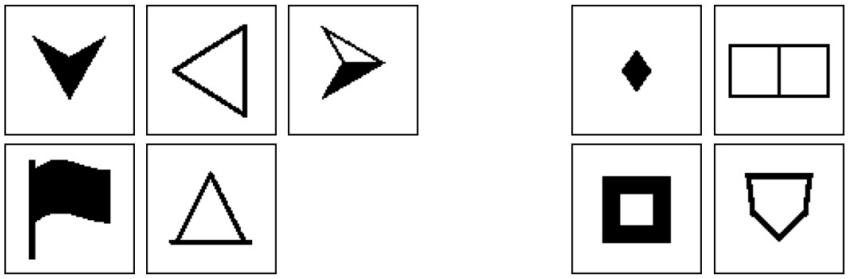


FIGURE A.42: Distribute three in closed fill style (full, empty, half-shaded). Answer=3.

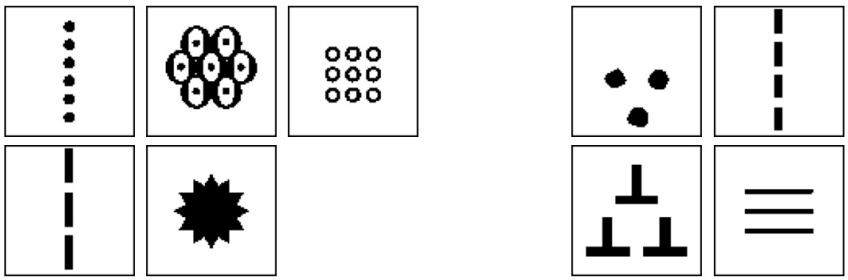


FIGURE A.43: Distribute three closure shapes (line, circle, square). Answer=4.

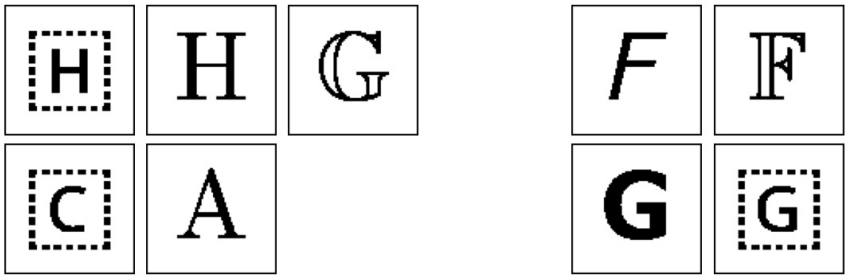


FIGURE A.44: Distribute three character style (dashed, thin, empty-bold). Answer=2.

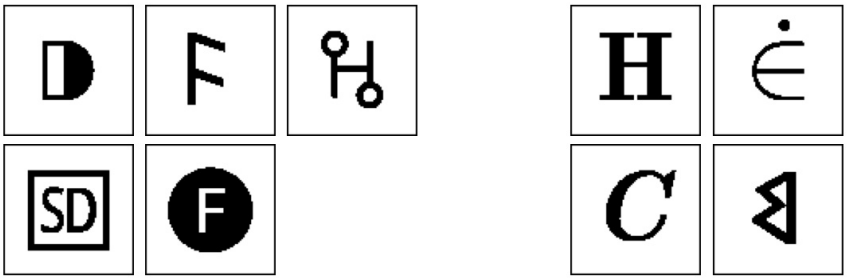


FIGURE A.45: Distribute three Latin characters (D, F, H). Answer=4.

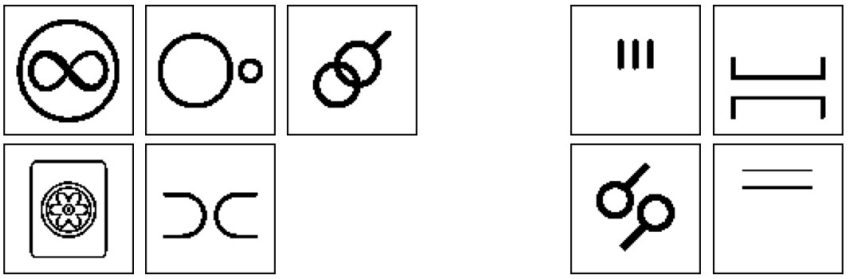


FIGURE A.46: Distribute three relational positions (centre, horizontal, diagonal). Answer=3.

FIGURE A.47: Distribute three kinds of relational rotation (parallel, towards, converging). Answer=2.

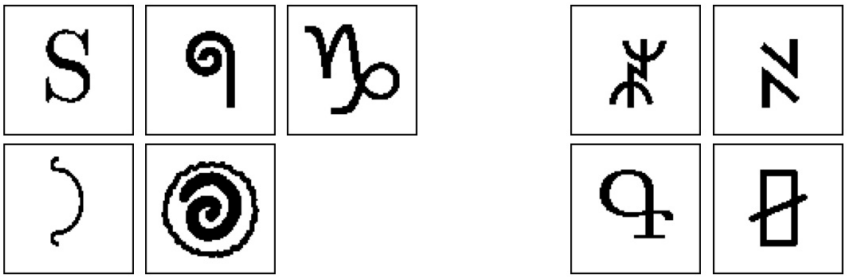


FIGURE A.48: Distribute three line stroke features (wiggle, spiral, loop). Answer=3.

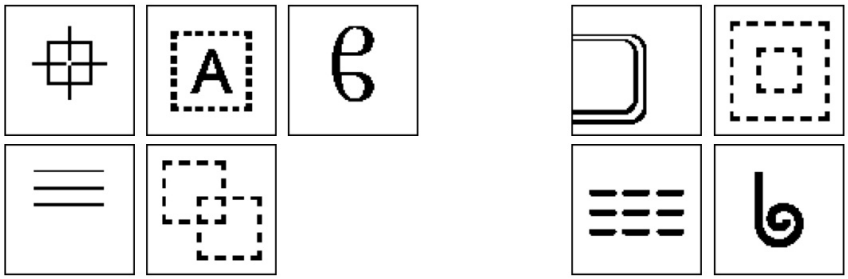


FIGURE A.49: Distribute three line rendering styles (thin, dashed, bold). Answer=4.

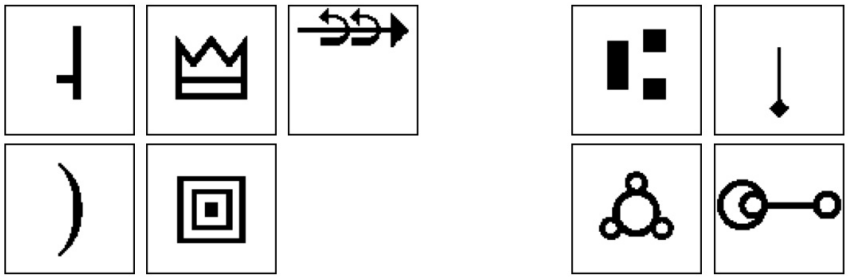


FIGURE A.50: Progression in aspect ratio (tall, square, wide). Answer=4.

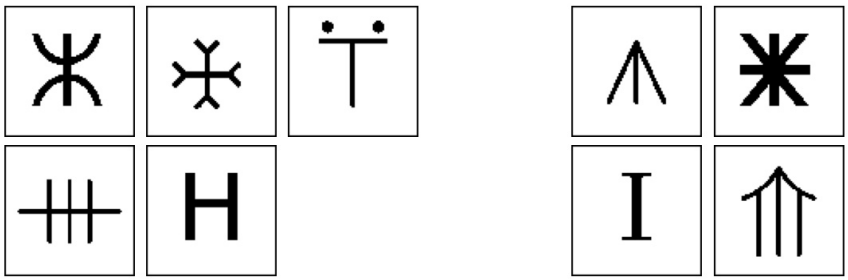


FIGURE A.51: Progression in number of base contacts (3, 2, 1). Answer=3.

FIGURE A.52: Progression in number of internal solid components (1, 2, 3). Answer=1.

FIGURE A.53: Progression in number of total solid components (2, 3, 4). Answer=4.

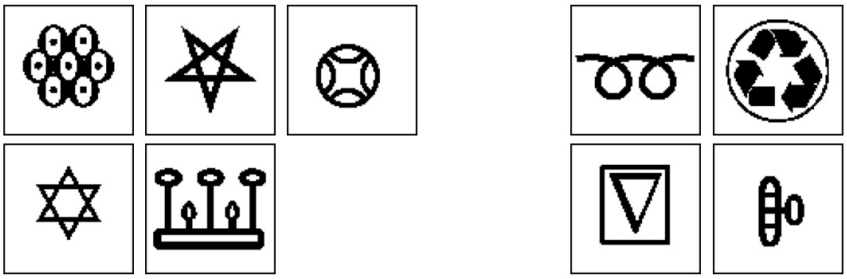


FIGURE A.54: Progression in number of internal spaces (7, 6, 5). Answer=4.

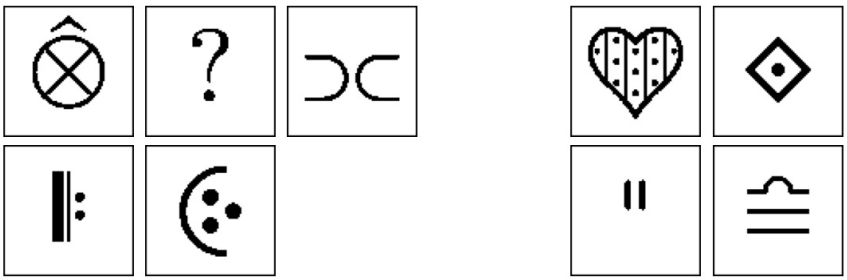


FIGURE A.55: Progression in number of unique solid components (3, 2, 1). Answer=3.

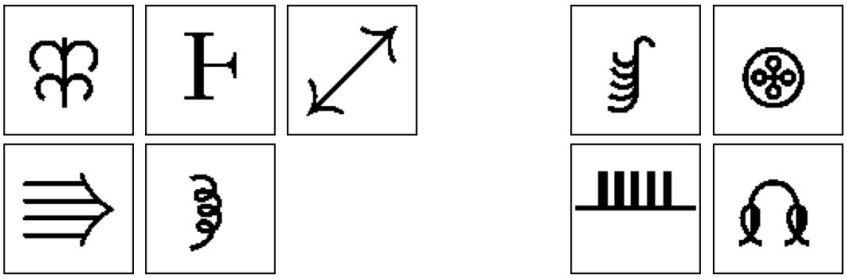


FIGURE A.56: Progression in number of connected components (4, 3, 2). Answer=4.

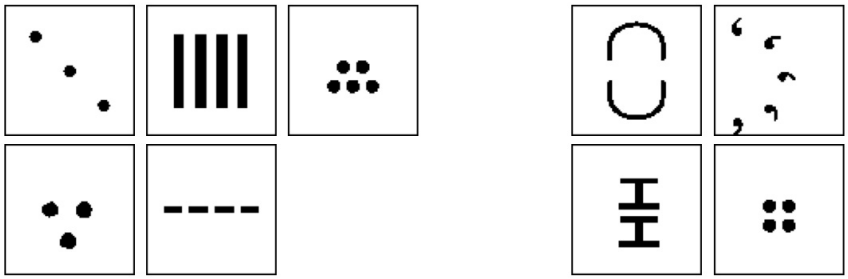


FIGURE A.57: Progression in number of disconnected components (3, 4, 5). Answer=2.

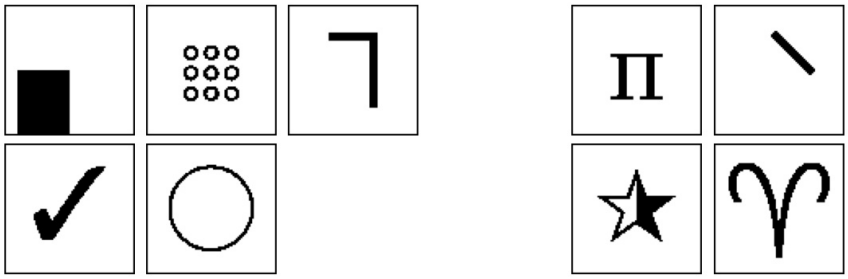


FIGURE A.58: Progression / movement in global mass centroid (South-West, Centre, North-East). Answer=2.

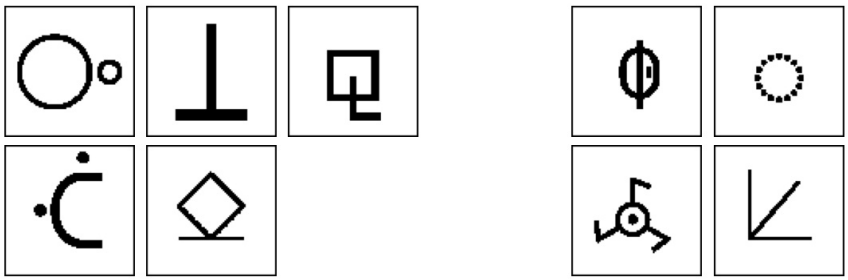


FIGURE A.59: Progression in interaction type (none, touching, overlap). Answer=1.

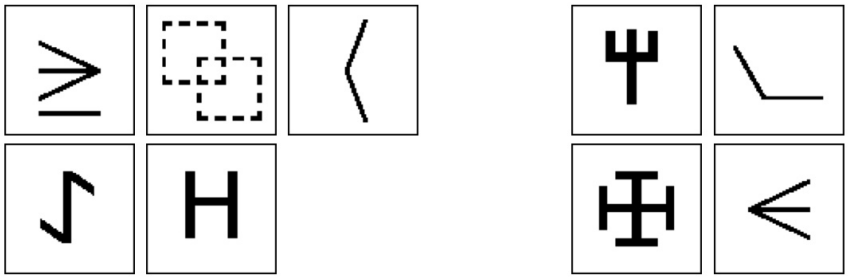


FIGURE A.60: Progression in angle of a contained intersection (acute, orthogonal, obtuse). Answer=2.

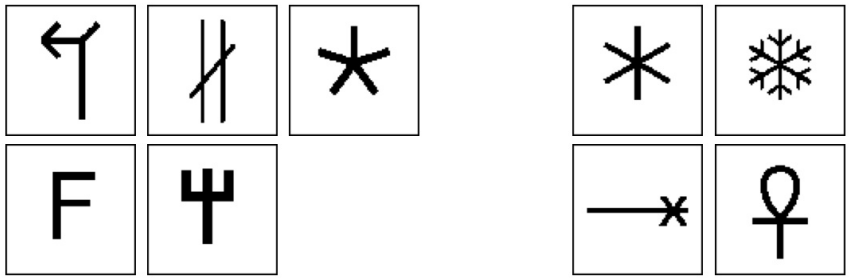


FIGURE A.61: Progression in number of emanating lines from a contained intersection (3, 4, 5). Answer=4.

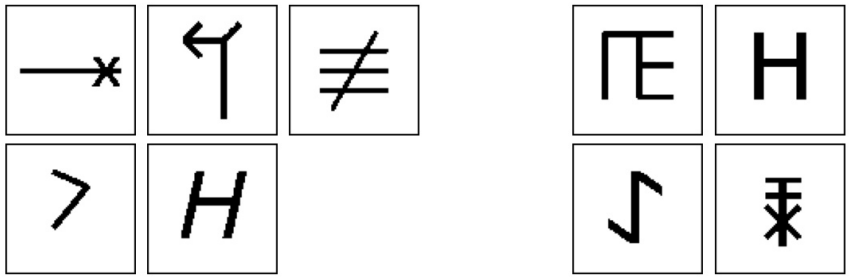


FIGURE A.62: Progression in number of intersections (1, 2, 3). Answer=4.

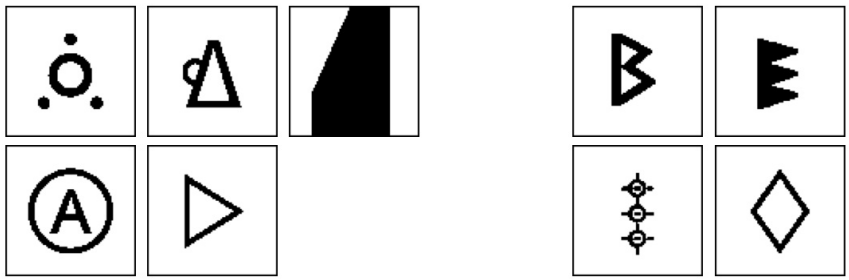


FIGURE A.63: Progression in number of shape sides (1, 3, 5). Answer=1.

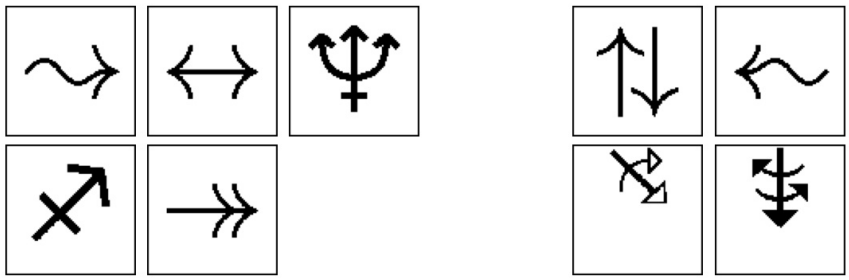


FIGURE A.64: Progression in number of arrows (1, 2, 3). Answer=4.

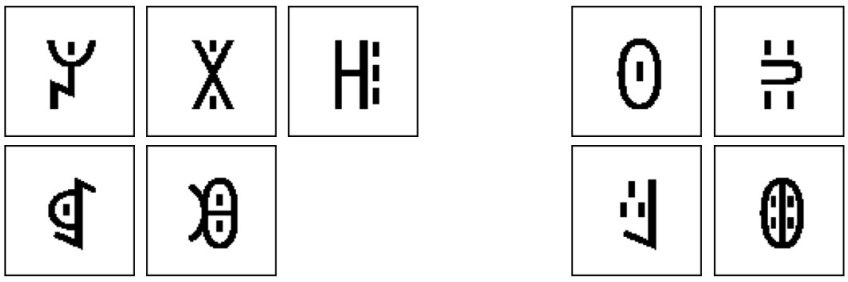


FIGURE A.65: Progression in number of dashes (1, 2, 3). Answer=3.

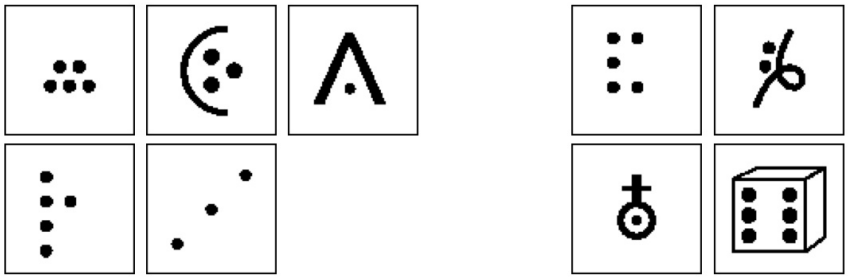


FIGURE A.66: Progression in number of dots (5, 3, 1). Answer=3.

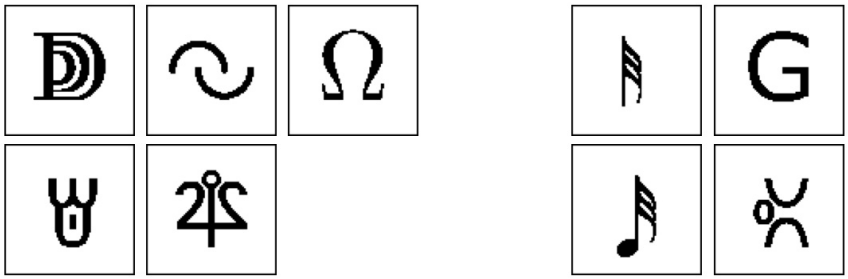


FIGURE A.67: Progression in number of curved strokes (3, 2, 1). Answer=2.

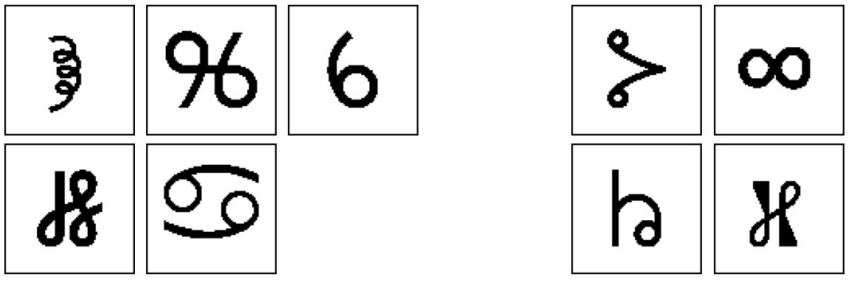


FIGURE A.68: Progression in number of loops (3, 2, 1). Answer=3.

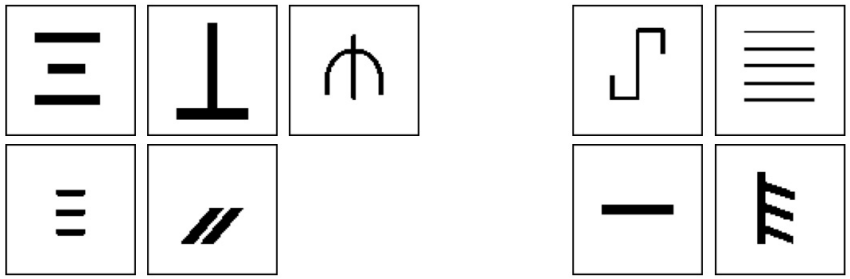


FIGURE A.69: Progression in number of straight lines (3, 2, 1). Answer=3.

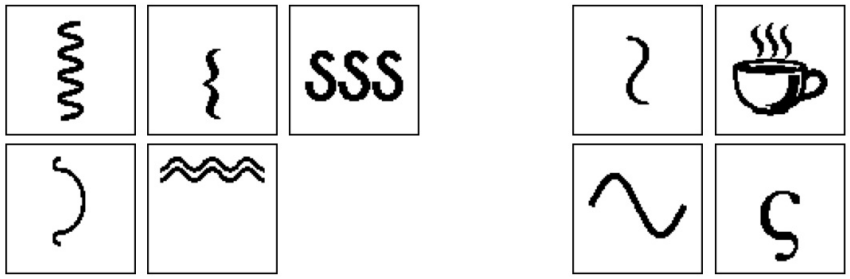


FIGURE A.70: Progression in number of "wiggly" lines (1, 2, 3). Answer=2.

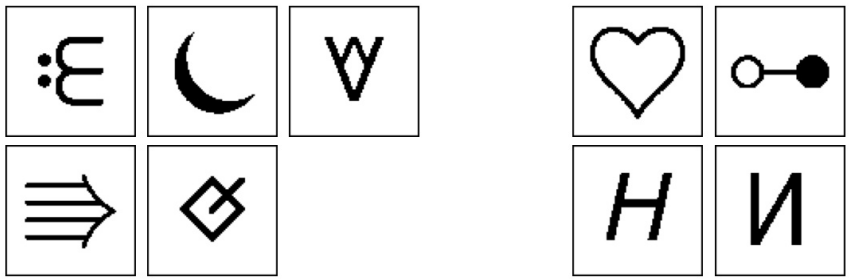


FIGURE A.71: Progression in symmetry angle (horizontal, diagonal, vertical). Answer=1.

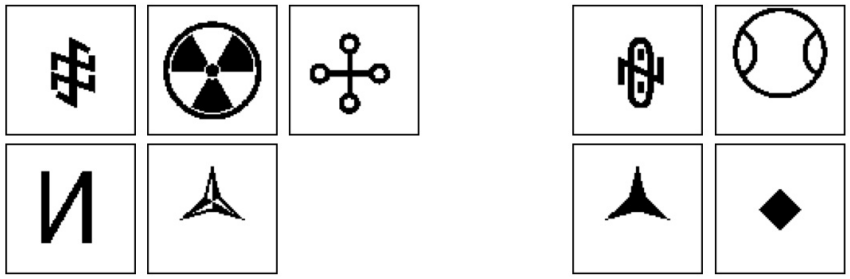


FIGURE A.72: Progression in degree of rotational symmetry (2, 3, 4). Answer=4.

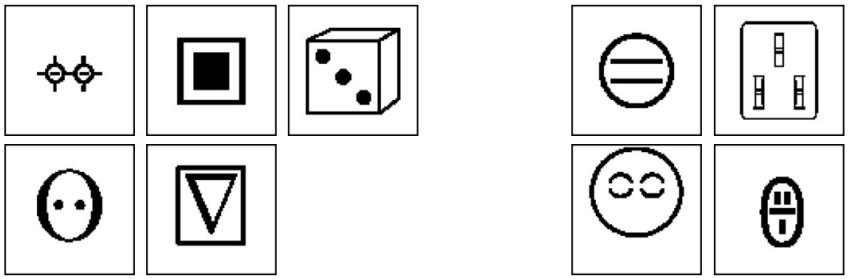


FIGURE A.73: Arithmetic in the number of internal solid components (2 + 1 = 3). Answer=2.

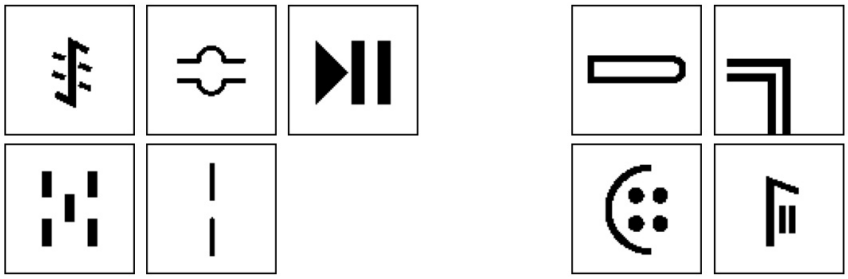


FIGURE A.74: Arithmetic in the number of total solid components (5 - 2 = 3). Answer=4.

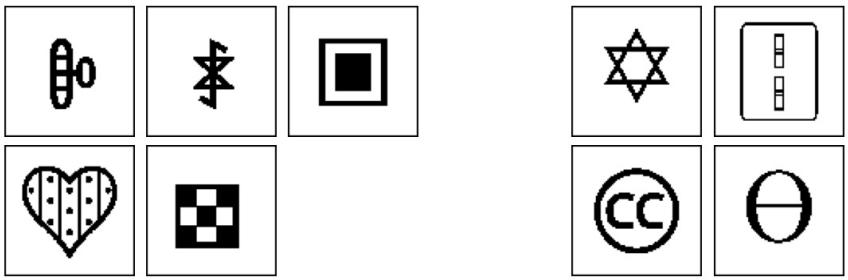


FIGURE A.75: Arithmetic in the number of internal spaces (5 - 4 = 1). Answer=3.

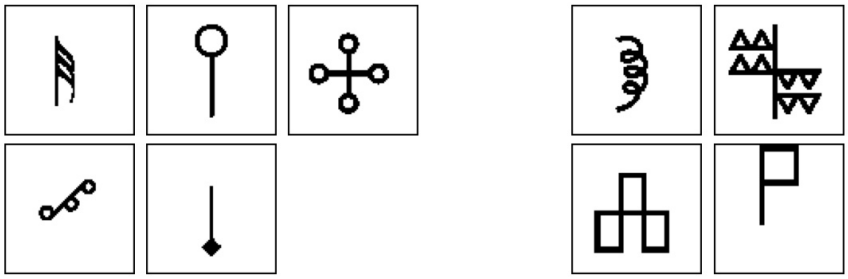


FIGURE A.76: Arithmetic in the number of connected elements (3 + 1 = 4). Answer=2.

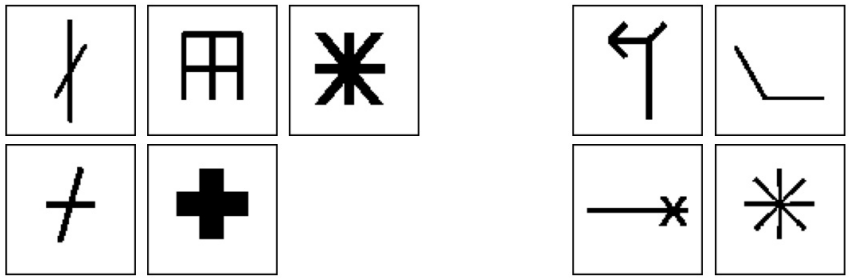


FIGURE A.77: Arithmetic in the number of emanating lines in intersections (4 + 4 = 8). Answer=4.

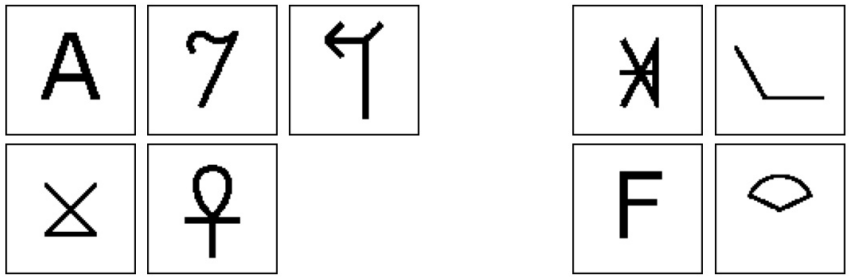


FIGURE A.78: Arithmetic in the number of intersections (3 - 1 = 2). Answer=3.

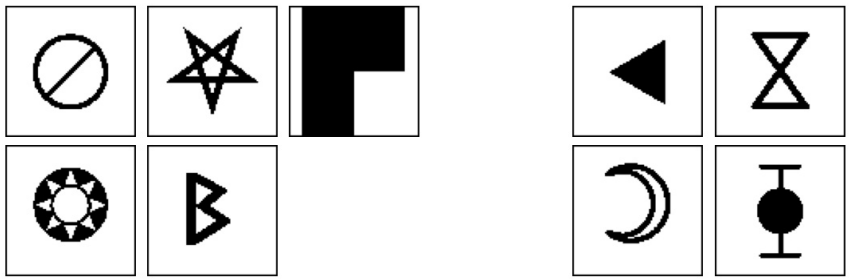


FIGURE A.79: Arithmetic in the number of shape sides (1 + 5 = 6). Answer=2.

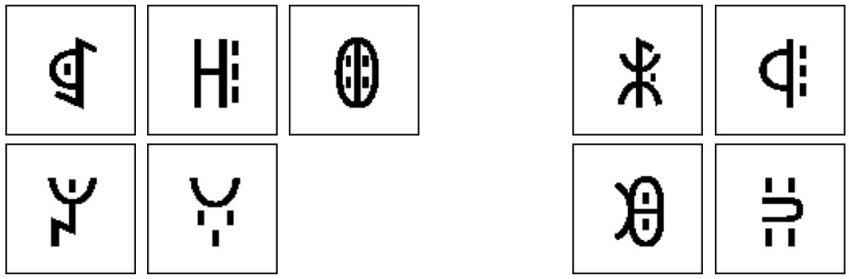


FIGURE A.80: Arithmetic in the number of dashes (1 + 3 = 4). Answer=4.

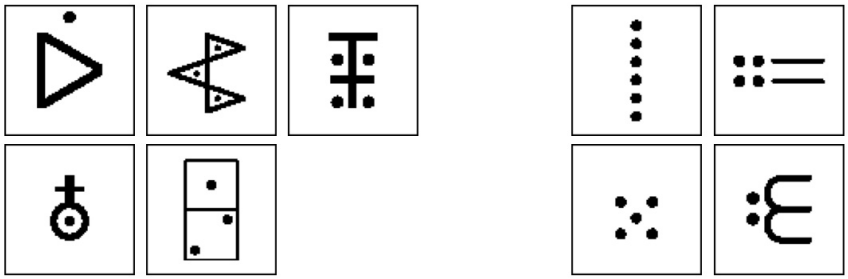


FIGURE A.81: Arithmetic in the number of dots (1 + 3 = 4). Answer=2.

FIGURE A.82: Arithmetic in the number of loops (1 + 1 = 2). Answer=3.

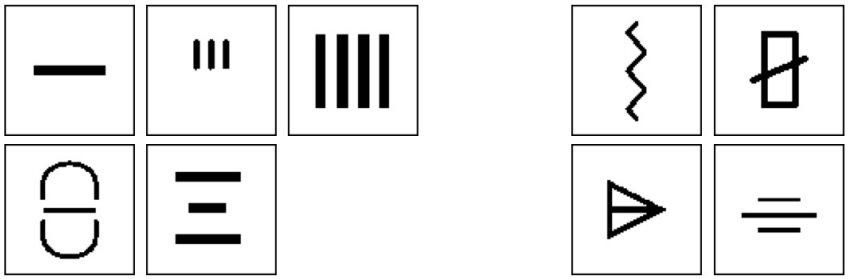


FIGURE A.83: Arithmetic in the number of straight lines (1 + 3 = 4). Answer=3.

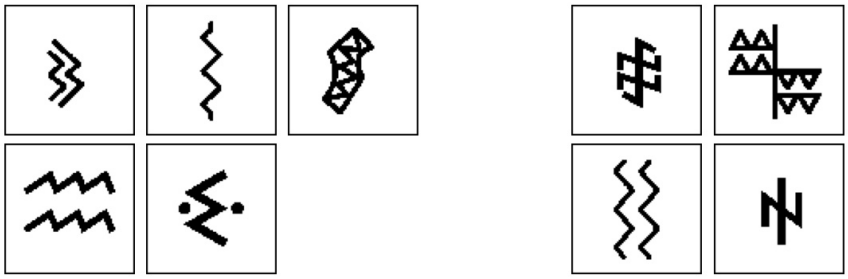


FIGURE A.84: Arithmetic in the number of zig-zag lines (2 - 1 = 1). Answer=4.

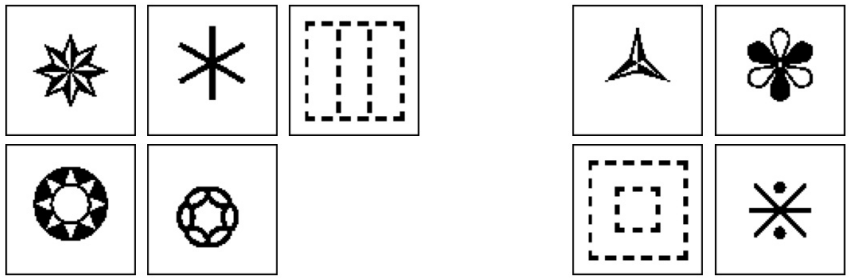


FIGURE A.85: Arithmetic in the degree of rotational symmetry (8 - 6 = 2). Answer=4.

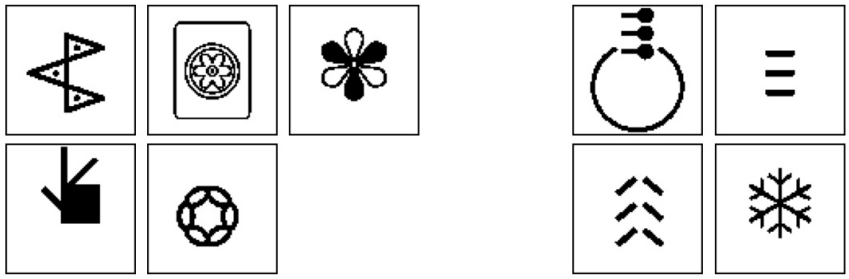


FIGURE A.86: Union in gestalt number (3, then 6, then both 3 and 6). Answer=3.

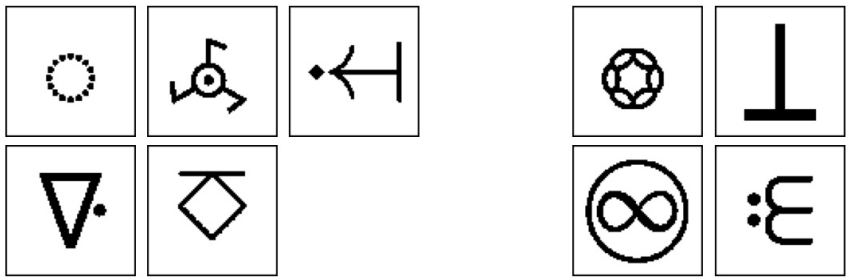


FIGURE A.87: Union in interaction type (none, then touching, then both none and touching). Answer=4.

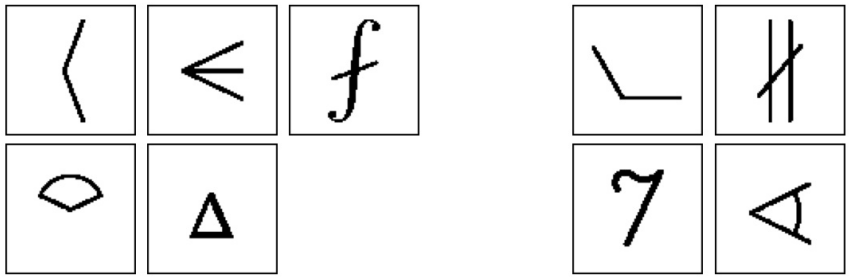


FIGURE A.88: Union in intersection angle (obtuse, then acute, then both obtuse and acute). Answer=2.

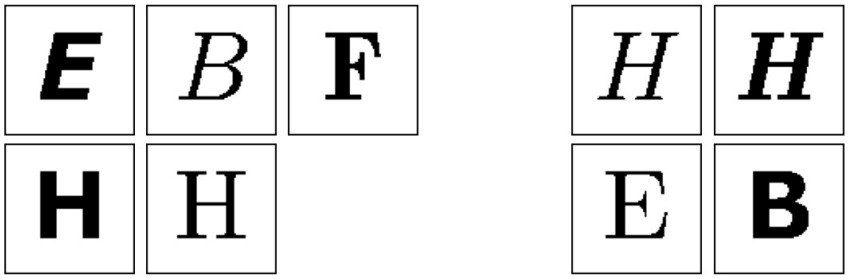


FIGURE A.89: Union in character style (bold, then serif, then both bold and serif). Answer=2.

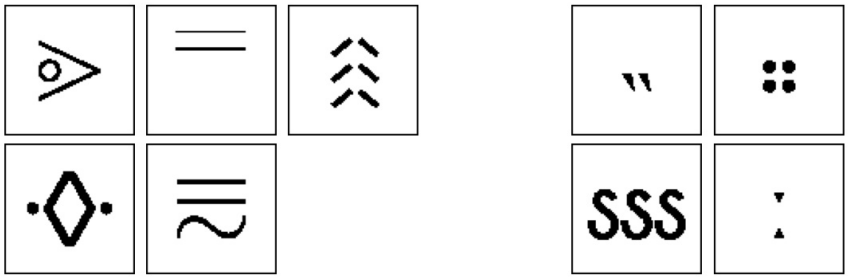


FIGURE A.90: Union in relational position (horizontal, then vertical, then both horizontal and vertical). Answer=2.

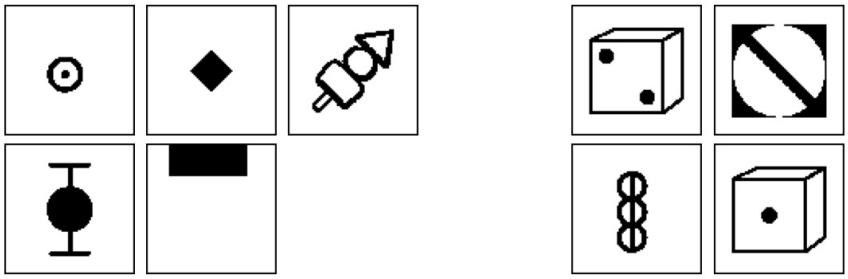


FIGURE A.91: Union in shape type (one-sided, then four-sided, then both one and four-sided). Answer=2.

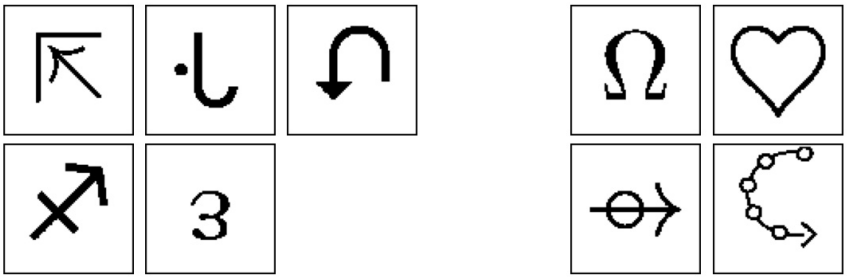


FIGURE A.92: Union in line stroke feature (arrow, then curve, then curved arrow). Answer=4.

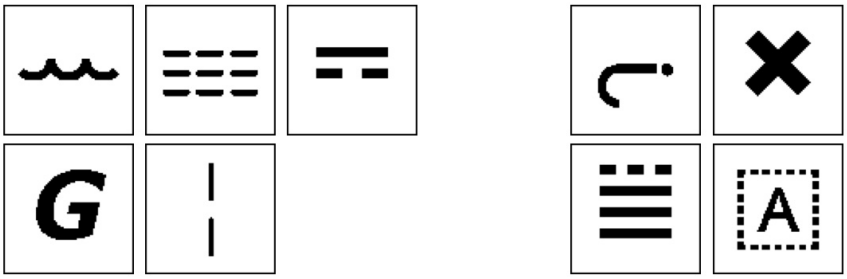


FIGURE A.93: Union in line stroke style (bold, then dashed, then bold-dashed). Answer=3.

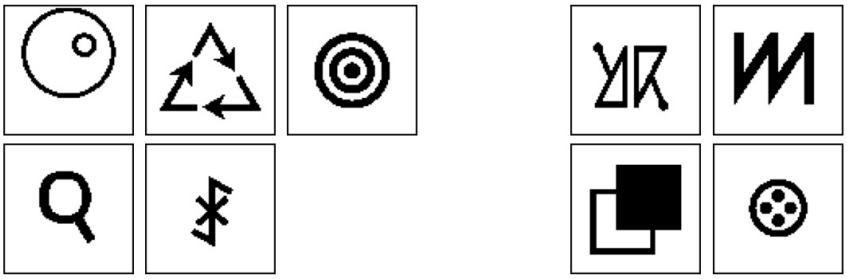


FIGURE A.94: Union in symmetry angle (diagonal, then rotational, then both diagonal and rotational lines). Answer=4.

Bibliography

[1] Saint Thomas Aquinas et al. *The summa theologica: Complete edition*. Catholic Way Publishing, 2014.

[2] B Ballaro, P Reas, and D Tegolo. "Elliptical Fourier Descriptors for shape retrieval in biological images". In: *International Conference on Electronics, Control & Signal Processing. SG*. 2002.

[3] Imon Banerjee et al. ""Shortcuts" causing bias in radiology artificial intelligence: causes, evaluation and mitigation." In: *Journal of the American College of Radiology* (2023).

[4] David Barrett et al. "Measuring abstract reasoning in neural networks". In: *International conference on machine learning*. PMLR. 2018, pp. 511–520.

[5] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[6] Yonatan Belinkov. "Probing classifiers: Promises, shortcomings, and advances". In: *Computational Linguistics* 48.1 (2022), pp. 207–219.

[7] Yaniv Benny, Niv Pekar, and Lior Wolf. "Scale-localized abstract reasoning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12557–12565.

[8] L. Bernstein. *The Unanswered Question: Six Talks at Harvard*. Charles Eliot Norton lectures v. 1-3. Harvard University Press, 1976. ISBN: 9780674920019. URL: https://books.google.com.au/books?id=30ITnKULvPIC.

[9] Eli Bingham et al. "Pyro: Deep Universal Probabilistic Programming". In: *Journal of Machine Learning Research* (2018).

[10] Rishi Bommasani et al. "On the opportunities and risks of foundation models". In: *arXiv preprint arXiv:2108.07258* (2021).

[11] M.M Bongard. *Pattern Recognition*. Spartan Books, 1967.

[12] George EP Box. "Science and statistics". In: *Journal of the American Statistical Association* (1976), pp. 791–799.

[13] Wieland Brendel and Matthias Bethge. "Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet". In: (Mar. 2019). arXiv: 1904.00760 [cs.CV].

[14] Darryl Bruce. "Fifty years since Lashley's In search of the Engram: refutations and conjectures". In: *Journal of the History of the Neurosciences* 10.3 (2001), pp. 308–318.

[15] Yuri Burda et al. "Large-Scale Study of Curiosity-Driven Learning". In: (Aug. 2018). arXiv: 1808.04355 [cs.LG].

[16] Christopher P Burgess et al. "Monet: Unsupervised scene decomposition and representation". In: *arXiv preprint arXiv:1901.11390* (2019).

[17] 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." In: *Psychological review* 97.3 (1990), p. 404.

[18] Daniel Casasanto. "All concepts are ad hoc concepts". In: *The conceptual mind: New directions in the study of the concepts*. MIT press. 2015, pp. 543–566.

[19] Raymond B Cattell. "Theory of fluid and crystallized intelligence: A critical experiment." In: *Journal of educational psychology* 54.1 (1963), p. 1.

[20] David J Chalmers. "On sense and intension". In: *Philosophical perspectives* 16 (2002), pp. 135–182.

[21] David J Chalmers, Robert M French, and Douglas R Hofstadter. "High-level perception, representation, and analogy: A critique of artificial intelligence methodology". In: *Journal of Experimental & Theoretical Artificial Intelligence* 4.3 (1992), pp. 185–211.

[22] William Charlton et al. *Aristotle's physics: Books I and II*. Oxford University Press, 1983.

[23] Tian Qi Chen et al. "Isolating sources of disentanglement in variational autoencoders". In: *Advances in Neural Information Processing Systems*. 2018, pp. 2610–2620.

[24] Anoop Cherian et al. "Are deep neural networks SMARTer than second graders?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 10834–10844.

[25] François Chollet. "On the measure of intelligence". In: *arXiv preprint arXiv:1911.01547* (2019).

[26] Jacob Cohen. "Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit." In: *Psychological bulletin* 70.4 (1968), p. 213.

[27] Eric Crawford and Joelle Pineau. "Spatially invariant unsupervised object detection with convolutional neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3412–3420.

[28] Grégoire Delétang et al. "Language modeling is compression". In: *arXiv preprint arXiv:2309.10668* (2023).

[29] G. Dicker. *Hume's Epistemology and Metaphysics: An Introduction*. Taylor & Francis, 2002. ISBN: 9781134714254. URL: https://books.google.com.au/books?id=lnGGAgAAQBAJ.

[30] David H Douglas and Thomas K Peucker. "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature". In: *Cartographica: the international journal for geographic information and geovisualization* 10.2 (1973), pp. 112–122.

[31] Dominik Maria Endres and Johannes E Schindelin. "A new metric for probability distributions". In: *IEEE Transactions on Information theory* 49.7 (2003), pp. 1858–1860.

[32] Martin Engelcke et al. "Genesis: Generative scene inference and sampling with object-centric latent representations". In: *arXiv preprint arXiv:1907.13052* (2019).

[33] SM Ali Eslami et al. "Attend, infer, repeat: Fast scene understanding with generative models". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3225–3233.

[34] Richard Evans et al. "Making sense of sensory input". In: (Oct. 2019). arXiv: 1910.02227 [cs.AI].

[35] Kuang Tih Fann. *Peirce's theory of abduction*. Springer Science & Business Media, 2012.

[36] Jacob Feldman. "How surprising is a simple pattern? Quantifying "Eureka!"" In: *Cognition* 93.3 (2004), pp. 199–224.

[37] Leon Festinger. *A Theory of Cognitive Dissonance*. Stanford University Press, 1957.

[38] Kenneth D Forbus, Dedre Gentner, and Keith Law. "MAC/FAC: A model of similarity-based retrieval". In: *Cognitive science* 19.2 (1995), pp. 141–205.

[39] Harry E Foundalis. "Unification of Clustering, Concept Formation, Categorization, and Analogy Making". Unpublished manuscript.

[40] Elias Frantar and Dan Alistarh. "Sparsegpt: Massive language models can be accurately pruned in one-shot". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 10323–10337.

[41] Gottlob Frege. "Über sinn und bedeutung". In: *Zeitschrift für Philosophie und philosophische Kritik* 100 (1892), pp. 25–50.

[42] Karl Friston. "The free-energy principle: a unified brain theory?" In: *Nature reviews neuroscience* 11.2 (2010), pp. 127–138.

[43] Karl Friston. "The history of the future of the Bayesian brain". In: *NeuroImage* 62.2 (2012), pp. 1230–1233.

[44] Alberto Garcia-Garcia et al. "A Review on Deep Learning Techniques Applied to Semantic Segmentation". In: (Apr. 2017). arXiv: 1704.06857 [cs.CV].

[45] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *arXiv preprint arXiv:1811.12231* (2018).

[46] Robert Geirhos et al. "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.

[47] Dedre Gentner. "Structure-mapping: A theoretical framework for analogy". In: *Cogn. Sci.* 7.2 (Apr. 1983), pp. 155–170.

[48] Dedre Gentner, Keith J Holyoak, and Boicho N Kokinov. *The analogical mind: Perspectives from cognitive science*. MIT press, 2001.

[49] Dedre Gentner and Arthur B Markman. "Structure mapping in analogy and similarity." In: *American psychologist* 52.1 (1997), p. 45.

[50] Edmund L Gettier. "Is justified true belief knowledge?" In: *analysis* 23.6 (1963), pp. 121–123.

[51] Mary L Gick and Keith J Holyoak. "Analogical problem solving". In: *Cognitive psychology* 12.3 (1980), pp. 306–355.

[52] Grant Gillett. "Representation, meaning, and thought". In: (1992).

[53] Jianping Gou et al. "Knowledge distillation: A survey". In: *International Journal of Computer Vision* 129 (2021), pp. 1789–1819.

[54] Klaus Greff et al. "Multi-object representation learning with iterative variational inference". In: *arXiv preprint arXiv:1903.00450* (2019).

[55] Shuyue Guan and Murray Loew. "Understanding the Ability of Deep Neural Networks to Count Connected Components in Images". In: *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*. IEEE. 2020, pp. 1–7.

[56] Michelle Guo et al. "Neural graph matching networks for fewshot 3d action recognition". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 653–669.

[57] Lukas Hahne et al. "Attention on Abstract Visual Reasoning". In: *arXiv preprint arXiv:1911.05990* (2019).

[58] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1024–1034.

[59] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[60] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. New York: John Wiley & Sons, 1949.

[61] Jose Hernandez-Orallo. "Beyond the Turing test". In: *Journal of Logic, Language and Information* 9 (2000), pp. 447–466.

[62] Irina Higgins et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." In: *Iclr* 2.5 (2017), p. 6.

[63] Irina Higgins et al. "Early Visual Concept Learning with Unsupervised Deep Learning". In: (June 2016). arXiv: 1606.05579 [stat.ML].

[64] Irina Higgins et al. "Towards a Definition of Disentangled Representations". In: (Dec. 2018). arXiv: 1812.02230 [cs.LG].

[65] Felix Hill et al. "Learning to Make Analogies by Contrasting Abstract Relational Structure". In: (Jan. 2019). arXiv: 1902.00120 [cs.AI].

[66] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the knowledge in a neural network". In: *arXiv preprint arXiv:1503.02531* (2015).

[67] D. Hofstadter and E. Sander. *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. Basic Books, 2013. ISBN: 9780465021581. URL: https://books.google.com.au/books?id=XkQT5eTnurYC.

[68] D.R. Hofstadter and Fluid Analogies Research Group. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Penguin Press Science Series. Penguin, 1998. ISBN: 9780140258356. URL: https://books.google.com.au/books?id=iit7QgAACAAJ.

[69] Douglas R Hofstadter. "Analogy as the core of cognition". In: *The analogical mind: Perspectives from cognitive science* (2001), pp. 499–538.

[70] Douglas R Hofstadter, Melanie Mitchell, and Robert Matthew French. *Fluid concepts and creative analogies: A theory and its computer implementation*. University of Michigan, Cognitive Science and Machine Intelligence Laboratory, 1987.

[71] Douglas R. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books Inc., 1979.

[72] Andreas Holzinger, Michael Kickmeier-Rust, and Heimo Müller. "KANDINSKY patterns as IQ-test for machine learning". In: *International cross-domain conference for machine learning and knowledge extraction*. Springer. 2019, pp. 1–14.

[73] Sheng Hu et al. "Stratified rule-aware network for abstract visual reasoning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2. 2021, pp. 1567–1574.

[74] John Hughes. "Why functional programming matters". In: *The computer journal* 32.2 (1989), pp. 98–107.

[75] Marcus Hutter. "On the foundations of universal sequence prediction". In: *International Conference on Theory and Applications of Models of Computation*. Springer. 2006, pp. 408–420.

[76] Frank Jackson. *From metaphysics to ethics: A defence of conceptual analysis*. Clarendon Press, 1998.

[77] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks". In: *Advances in neural information processing systems*. 2015, pp. 2017–2025.

[78] Frederick Jelinek. "Some of My Best Friends Are Linguists". In: *Language Resources and Evaluation* 39.1 (2005), pp. 25–34. ISSN: 1574020X, 15728412. URL: http://www.jstor.org/stable/30200539 (visited on 01/26/2024).

[79] Petter Johansson et al. "Choice blindness and preference change: you will like this paper better if you (believe you) chose to read it!" In: *Journal of Behavioral Decision Making* 27.3 (2014), pp. 281–289.

[80] Justin Johnson et al. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2901–2910.

[81] Friston Karl. "A Free Energy Principle for Biological Systems". en. In: *Entropy* 14.11 (Nov. 2012), pp. 2100–2121.

[82] Harmanpreet Kaur et al. "Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning". In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–14.

[83] Shaiyan Keshvari and Ruth Rosenholtz. "Pooling of continuous features provides a unifying account of crowding". In: *Journal of Vision* 16.3 (2016), pp. 39–39.

[84] Been Kim et al. "Do neural networks show gestalt phenomena? an exploration of the law of closure". In: *arXiv preprint arXiv:1903.01069* 2.8 (2019).

[85] Hyunjik Kim and Andriy Mnih. "Disentangling by Factorising". In: *International Conference on Machine Learning*. 2018, pp. 2649–2658.

[86] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[87] Thomas N Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: (Sept. 2016). arXiv: 1609.02907 [cs.LG].

[88] John R Kirby and Michael J Lawson. "Effects of strategy training on progressive matrices performance". In: *Contemporary Educational Psychology* 8.2 (1983), pp. 127–140.

[89] Małgorzata Kisielewska, Mariusz Urbański, and Katarzyna Paluszkiewicz. "Abduction in one intelligence test. Types of reasoning involved in solving Raven's Advanced Progressive Matrices". In: *Model-Based Reasoning in Science and Technology: Logical, Epistemological, and Cognitive Issues*. Springer. 2016, pp. 419–435.

[90] Jack Koch et al. "Objective robustness in deep reinforcement learning". In: *arXiv preprint arXiv:2105.14111* (2021).

[91] Saul A Kripke. "Naming and necessity: Lectures given to the princeton university philosophy colloquium". In: *Semantics of natural language*. Springer, 1980, pp. 253–355.

[92] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* 350.6266 (2015), pp. 1332–1338.

[93] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "The Omniglot challenge: a 3-year progress report". In: *Current Opinion in Behavioral Sciences* 29 (2019), pp. 97–104.

[94] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[95] Shane Legg and Marcus Hutter. "A Collection of Definitions of Intelligence". In: (June 2007). arXiv: 0706.3639 [cs.AI].

[96] Kenneth Li et al. "Emergent world representations: Exploring a sequence model trained on a synthetic task". In: *arXiv preprint arXiv:2210.13382* (2022).

[97] Yu Liang et al. "Explaining the black-box model: A survey of local interpretation methods for deep neural networks". In: *Neurocomputing* 419 (2021), pp. 168–182.

[98] Jianhua Lin. "Divergence measures based on the Shannon entropy". In: *IEEE Transactions on Information theory* 37.1 (1991), pp. 145–151.

[99] Alexandre Linhares. "A glimpse at the metaphysics of Bongard problems". In: *Artificial Intelligence* 121.1-2 (2000), pp. 251–270.

[100] Daniel R Little, Stephan Lewandowsky, and Thomas L Griffiths. "A bayesian model of rule induction in raven's progressive matrices". In: *Proceedings of the annual meeting of the cognitive science society*. Vol. 34. 34. 2012.

[101] Francesco Locatello et al. "Object-centric learning with slot attention". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11525–11538.

[102] Andrew Lovett and Kenneth Forbus. "Modeling visual problem solving as analogical reasoning." In: *Psychological review* 124.1 (2017), p. 60.

[103] Norman RF Maier. "Reasoning in humans. II. The solution of a problem and its appearance in consciousness." In: *Journal of comparative Psychology* 12.2 (1931), p. 181.

[104] Mikołaj Małkiński and Jacek Mańdziuk. "Deep Learning Methods for Abstract Visual Reasoning: A Survey on Raven's Progressive Matrices". In: *arXiv preprint arXiv:2201.12382* (2022).

[105] Gary Marcus. "Deep learning: A critical appraisal". In: *arXiv preprint arXiv:1801.00631* (2018).

[106] Linda BL Vodegel Matzen, Maurits W Van der Molen, and Ad CM Dudink. "Error analysis of Raven test performance". In: *Personality and Individual Differences* 16.3 (1994), pp. 433–445.

[107] P Mayring. "Qualitative Content Analysis Philipp Mayring 3. Basic Ideas of Content Analysis". In: *Forum Qualitative Sozialforschung*. Vol. 1. 10. 2000.

[108] John McCarthy et al. "A proposal for the Dartmouth summer research project on artificial intelligence (1955)". In: *Reprinted online at http://www-formal. stanford. edu/jmc/history/dartmouth/dartmouth. html* (2018).

[109] Michael McCloskey and Neal J Cohen. "Catastrophic interference in connectionist networks: The sequential learning problem". In: *Psychology of learning and motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[110] Thomas McGrath et al. "Acquisition of chess knowledge in alphazero". In: *arXiv preprint arXiv:2111.09259* (2021).

[111] S.B. McGrayne. *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of C*. Matematicas (E-libro). Yale University Press, 2011. ISBN: 9780300175097. URL: https://books.google.com.au/books?id=_Kx5xVGuLRIC.

[112] L.B. Meyer. *Emotion and Meaning in Music*. Jeff borrow list. University of Chicago Press, 1961. ISBN: 9780226521398. URL: https://books.google.com.au/books?id=HuWCVGKhwy0C.

[113] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial intelligence* 267 (2019), pp. 1–38.

[114] Chris Mingard et al. "Is SGD a Bayesian sampler? Well, almost". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 3579–3642.

[115] Chris Mingard et al. "Neural networks are a priori biased towards boolean functions with low entropy". In: *arXiv preprint arXiv:1909.11522* (2019).

[116] Shanka Subhra Mondal, Taylor Webb, and Jonathan D Cohen. "Learning to reason over visual objects". In: *arXiv preprint arXiv:2303.02260* (2023).

[117] Weili Nie et al. "Bongard-logo: A new benchmark for human-level concept learning and reasoning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16468–16480.

[118] Bruce Nielson and Daniel C Elton. "Induction, Popper, and machine learning". In: *arXiv preprint arXiv:2110.00840* (2021).

[119] F. Nietzsche, O. Levy, and J.M. Kennedy. *Homer and Classical Philology*. Floating Press, 2013. ISBN: 9781776527182. URL: https://books.google.com.au/books?id=ImmsDgAAQBAJ.

[120] Victor Vikram Odouard and Melanie Mitchell. "Evaluating Understanding on Conceptual Abstraction Benchmarks". In: *arXiv preprint arXiv:2206.14187* (2022).

[121] Adam Paszke et al. "Automatic differentiation in PyTorch". In: *NIPS-W*. 2017.

[122] Jonathan W Peirce. "PsychoPy—psychophysics software in Python". In: *Journal of neuroscience methods* 162.1-2 (2007), pp. 8–13.

[123] Joshua C Peterson, Dawn Chen, and Thomas L Griffiths. "Parallelograms revisited: Exploring the limitations of vector space models for simple analogies". In: *Cognition* 205 (2020), p. 104440.

[124] Karl R Popper. "The logic of scientific discovery". In: *Central Works of Philosophy v4: Twentieth Century: Moore to Popper* 4 (2015), p. 262.

[125] Hilary Putnam. "Meaning and Reference". In: *Journal of Philosophy* 70.19 (1973), pp. 699–711. DOI: 10.2307/2025079.

[126] Yonggang Qi et al. "PQA: Perceptual Question Answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12056–12064.

[127] John Raven. "The Raven's progressive matrices: change and stability over culture and time". In: *Cognitive psychology* 41.1 (2000), pp. 1–48.

[128] John C Raven and JH Court. *Raven's progressive matrices*. Western Psychological Services Los Angeles, CA, 1938.

[129] Ali Razavi et al. "Preventing Posterior Collapse with delta-VAEs". In: (Jan. 2019). arXiv: 1901.03416 [cs.LG].

[130] Scott Reed et al. "A generalist agent". In: *arXiv preprint arXiv:2205.06175* (2022).

[131] Henry L Roediger and Kurt A DeSoto. "Psychology of reconstructive memory". In: *International encyclopedia of the social & behavioral sciences* 20.2 (2015), pp. 50–55.

[132] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[133] Drew Roselli, Jeanna Matthews, and Nisha Talagala. "Managing bias in AI". In: *Companion Proceedings of The 2019 World Wide Web Conference*. 2019, pp. 539–544.

[134] Lee Ross, David Greene, and Pamela House. "The "false consensus effect": An egocentric bias in social perception and attribution processes". In: *Journal of experimental social psychology* 13.3 (1977), pp. 279–301.

[135] J.J. Rousseau and J.T. Scott. *Essay on the Origin of Languages and Writings Related to Music*. Collected Writings of Rousseau Series. University Press of New England, 2009. ISBN: 9781611681277. URL: https://books.google.com.au/books?id=RYy9AgAAQBAJ.

[136] Robert D Rupert. "Embodied knowledge, conceptual change, and the a priori; or, justification, revision, and the ways life could go". In: *American Philosophical Quarterly* (2016), pp. 169–192.

[137] Adam Santoro et al. "A simple neural network module for relational reasoning". In: (June 2017). arXiv: 1706.01427 [cs.CL].

[138] Michael Schlichtkrull et al. "Modeling relational data with graph convolutional networks". In: *European Semantic Web Conference*. Springer. 2018, pp. 593–607.

[139] Jurgen Schmidhuber. "Simple algorithmic theory of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes". In: *Journal of SICE* 48.1 (2009).

[140] Laura Schroeter. *Gruesome diagonals*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2003.

[141] Gerhard Schurz. "Patterns of abduction". In: *Synthese* 164 (2008), pp. 201–234.

[142] John Searle. "Minds, Brains and Programs". In: *The Mind's I. New York: Basic Books* (2000).

[143] Thomas B Sebastian and Benjamin B Kimia. "Curves vs. skeletons in object recognition". In: *Signal Processing* 85.2 (Feb. 2005), pp. 247–263.

[144] A. Seth. *Being You: A New Science of Consciousness*. Faber & Faber, 2021. ISBN: 9780571337705. URL: https://books.google.com.au/books?id=7d9UygEACAAJ.

[145] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[146] Elliott Sober. *Ockam's Razor: A User's Manual*. 2015.

[147] Ray J Solomonoff. "A formal theory of inductive inference. Part I". In: *Information and control* 7.1 (1964), pp. 1–22.

[148] Charles Spearman. "General Intelligence, Objectively Determined and Measured". In: *The American Journal of Psychology* (1904).

[149] Aleksandar Stanić and Jürgen Schmidhuber. "R-SQAIR: Relational Sequential Attend, Infer, Repeat". In: *arXiv preprint arXiv:1910.05231* (2019).

[150] Samuel Stanton et al. "Does knowledge distillation really work?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6906–6919.

[151] Xander Steenbrugge et al. "Improving generalization for abstract reasoning tasks using disentangled feature representations". In: *arXiv preprint arXiv:1811.04784* (2018).

[152] Sjoerd van Steenkiste et al. "Are Disentangled Representations Helpful for Abstract Visual Reasoning?" In: (May 2019). arXiv: 1905.12506 [cs.LG].

[153] Sjoerd van Steenkiste et al. "Are Disentangled Representations Helpful for Abstract Visual Reasoning?" In: *Advances in Neural Information Processing Systems*. 2019, pp. 14222–14235.

[154] J Ridley Stroop. "Studies of interference in serial verbal reactions." In: *Journal of experimental psychology* 18.6 (1935), p. 643.

[155] Richard Sutton. "The bitter lesson". In: *Incomplete Ideas (blog)* 13 (2019), p. 12.

[156] Joshua Brett Tenenbaum. "A Bayesian framework for concept learning". PhD thesis. Massachusetts Institute of Technology, 1999.

[157] David M Thissen. "Information in wrong responses to the Raven Progressive Matrices". In: *Journal of Educational Measurement* (1976), pp. 201–214.

[158] Alan Mathison Turing et al. "On computable numbers, with an application to the Entscheidungsproblem". In: *J. of Math* 58.345-363 (1936), p. 5.

[159] Tomer D Ullman and Joshua B Tenenbaum. "Bayesian models of conceptual development: Learning as building models of the world". In: *Annual Review of Developmental Psychology* 2 (2020), pp. 533–558.

[160] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[161] Dorothy Walsh. "Occam's razor: A principle of intellectual elegance". In: *American Philosophical Quarterly* 16.3 (1979), pp. 241–244.

[162] Duo Wang, Mateja Jamnik, and Pietro Lio. "Unsupervised and interpretable scene discovery with Discrete-Attend-Infer-Repeat". In: *arXiv preprint arXiv:1903.06581* (2019).

[163] Ian Watson and Farhi Marir. "Case-based reasoning: A review". In: *The knowledge engineering review* 9.4 (1994), pp. 327–354.

[164] Taylor Webb, Keith J Holyoak, and Hongjing Lu. "Emergent analogical reasoning in large language models". In: *Nature Human Behaviour* 7.9 (2023), pp. 1526–1541.

[165] Taylor W Webb et al. "The relational bottleneck as an inductive bias for efficient abstraction". In: *arXiv preprint arXiv:2309.06629* (2023).

[166] Gerald Westheimer. "Was Helmholtz a Bayesian?" In: *Perception* 37.5 (2008), pp. 642–650.

[167] Timothy Williamson. *The philosophy of philosophy*. John Wiley & Sons, 2021.

[168] David H Wolpert and William G Macready. "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

[169] Yuhuai Wu et al. "The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning". In: *arXiv preprint arXiv:2007.04212* (2020).

[170] Jingyi Xu et al. "Abstract Visual Reasoning: An Algebraic Approach for Solving Raven's Progressive Matrices". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 6715–6724.

[171] Jianwei Yang et al. "Graph r-cnn for scene graph generation". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 670–685.

[172] Chi Zhang et al. "Abstract spatial-temporal reasoning via probabilistic abduction and execution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9736–9746.

[173] Chi Zhang et al. "Learning perceptual inference by contrasting". In: *Advances in Neural Information Processing Systems*. 2019, pp. 1073–1085.

[174] Chi Zhang et al. "Raven: A dataset for relational and analogical visual reasoning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5317–5327.

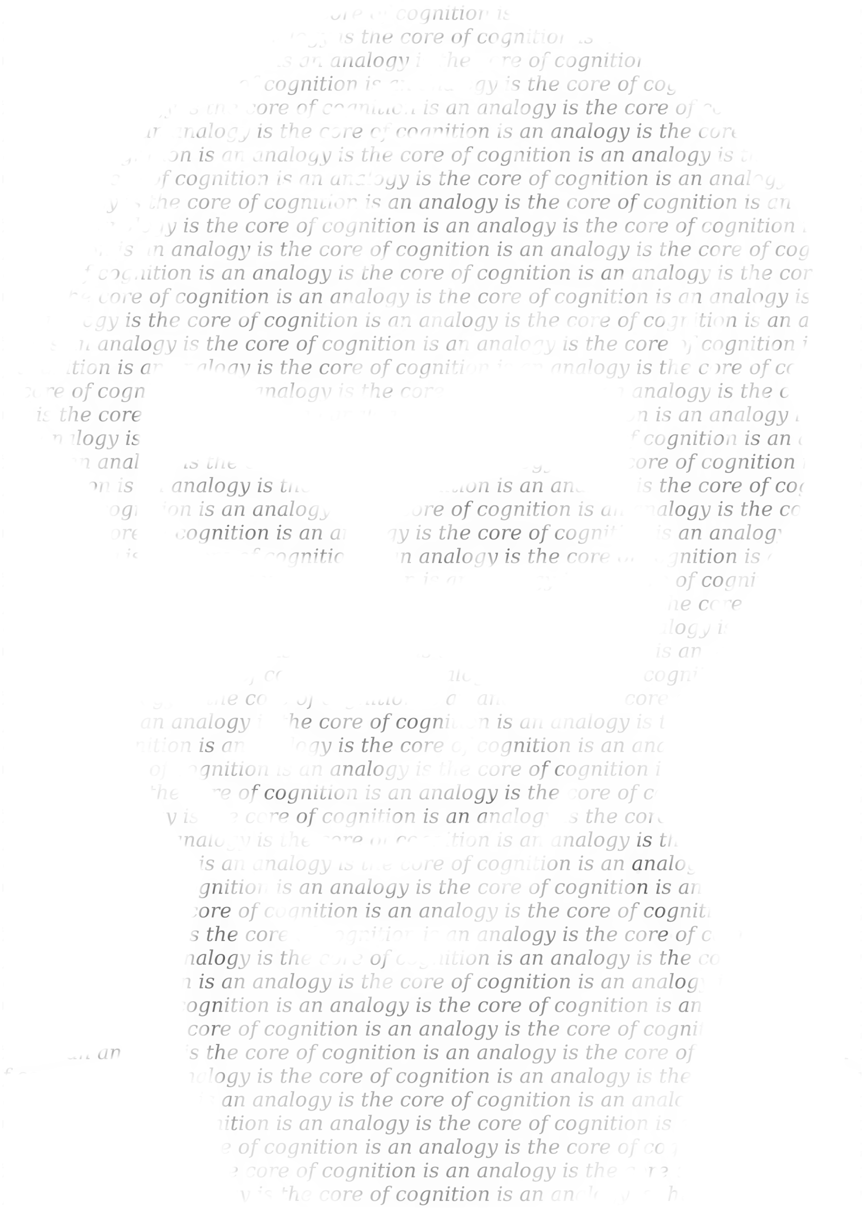
[175] Yu Zhang et al. "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742.

[176] Zhong-Qiu Zhao et al. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[177] Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. "Abstract Reasoning with Distracting Features". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5834–5845.

[178] Zhenzhu Zheng and Xi Peng. "Self-guidance: improve deep neural network generalization via knowledge distillation". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 3203–3212.

[179] Tao Zhuo and Mohan Kankanhalli. "Solving Raven's Progressive Matrices with Neural Networks". In: *arXiv preprint arXiv:2002.01646* (2020).


analogy is the core of cognition is an analogy is the core of cognition