A Closer Look at Generalisation in RAVEN

Steven Spratley, Krista Ehinger, and Tim Miller

School of Computing and Information Systems
The University of Melbourne, Victoria, Australia

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.

Keywords: visual reasoning, representation learning, scene understanding, raven's progressive matrices

1 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 [17]. 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 [7, 16], and has been considered to lie at the "core of cognition" [13]. 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) [19] 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 [20] and RAVEN [28] – 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 [2, 8, 6]. 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 paper, 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 [28]. 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.

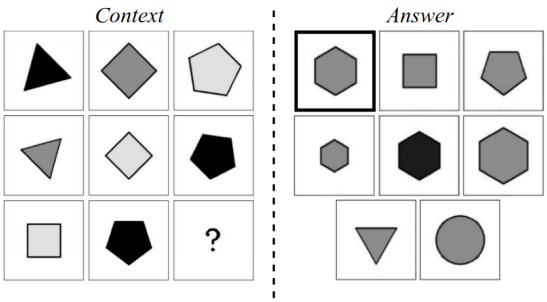

Fig. 1. An example RPM problem in RAVEN. In the context, the first two rows each have objects of a set size, of a progressively increasing number of sides, and with one of each colour. Therefore, the emboldened answer frame is correct; when inserted into the context, it allows the third row to adhere to the rules.

2 Background and Related Work

2.1 Raven's Progressive Matrices and Neural Networks

In the field of human intelligence testing, Raven's Progressive Matrices (RPMs) [19] 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 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 [20] and RAVEN [28]. Neither of these datasets are 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 paper's investigation.

The neural baselines introduced in these papers [20, 28] are both variations on the ResNet architecture [10], 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 [20] – 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 [28]. 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.* [29], 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.* [30] 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.* [9] substitute a more expressive Transformer network [25] 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 [31] 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 paper, 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 [13], 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.

2.2 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 [11], and is demonstrated to aid abstract reasoning tasks [24] and improve performance on PGM [23].

In the disentanglement literature, methods based on variational auto-encoders (VAEs) are ubiquitous [12, 15, 3], usually aiming to maximise the evidence lower bound (ELBO), $\mathcal{L}(\theta, \phi)$:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(z)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \tag{1}$$

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

$$p_\theta(x) = \int p_\theta(x|z)p(z)dz \tag{2}$$

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

$$\log p_\theta(x) = \mathcal{L}(\theta, \phi) + KL(q_\phi(z|x)||p(z|x)) \tag{3}$$

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 [2, 8, 5, 6]. 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) [6] 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 [14] (*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

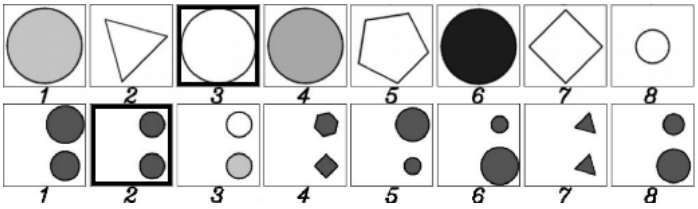

Fig. 2. Two example answer sets from problems in RAVEN. We can derive the correct answer (emboldened) from each set by finding the intersection of the set's modes of shape, colour, and scale factors. Essentially, "which frame has the most common features?"

object attended to. We seek to leverage these abilities of AIR as a preprocessing step over the RAVEN dataset.

3 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 [31]; they instead pre-train using Imagenet to help mitigate such overfitting. Upon rectifying 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." [28]

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 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 4), 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 ([30], benchmarking WReN in [29]) should still be correct, as their architectures already enforce this independent processing. Unfortunately, in [29], the model-level contrast summarizes common features within the answer set, and therefore misses this independence requirement. [31] also follows the methodology of [28]. This is of critical importance for the ongoing use of this dataset.

4 Architectures

In this section, we detail the three architectures benchmarked in this paper. The purpose of our ResNet model is to serve as an analogue to the original in [28], 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.

4.1 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 [20], 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.

4.2 Frame-relational ResNet (Rel-Base)

Improving on the baseline, Rel-Base encodes problems in two stages. The 4-layer encoder used in Section 4.1 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 [20], 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.

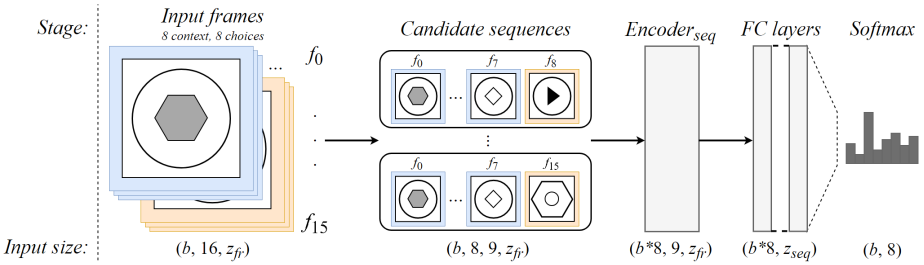

Fig. 3. Diagram of the basic method. Given a batch of b problems, b*8 candidate sequences are formed, independently encoded, and scored. For Rel-Base and Rel-AIR, frame embeddings of size z_{fr} are generated by additional stages. For ResNet, raw frames are used.

4.3 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 [6], which provides 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 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.

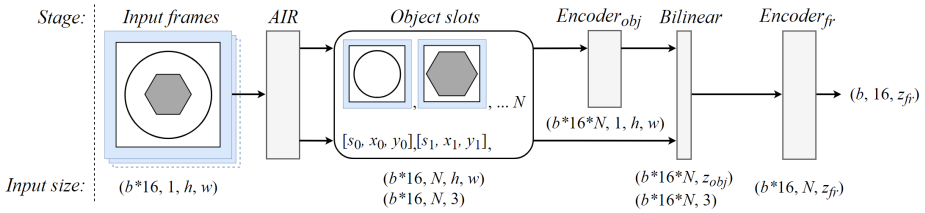

Fig. 4. Frame encoding in Rel-AIR. The AIR stage decomposes frames into a maximum N constituent objects and their associated scales and x,y positions; s_n, x_n, y_n. Second and third, each object is embedded (size z_{obj}), and processed via a bilinear layer to incorporate latent data. Finally, each frame's object embeddings are convolved together, resulting in overall frame embeddings.

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 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.

5 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, only the WReN [29] and LEN [30] 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 [18] and Pyro [1].

5.1 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 5), before fitting models to individual configurations. We do not make use of the provided auxiliary information, we restrict image

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

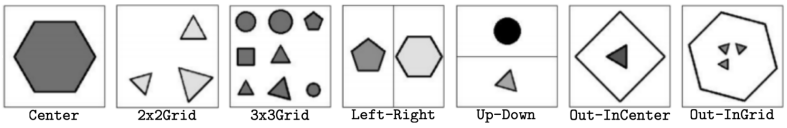

Fig. 5. Example frames from RAVEN's diverse problem configurations.

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 [29].

5.2 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 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 [9, 30], 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 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.

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 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 paper. Yet, between both PGM sets, this strongly suggests that no utility is lost in the simpler architecture of Rel-Base.

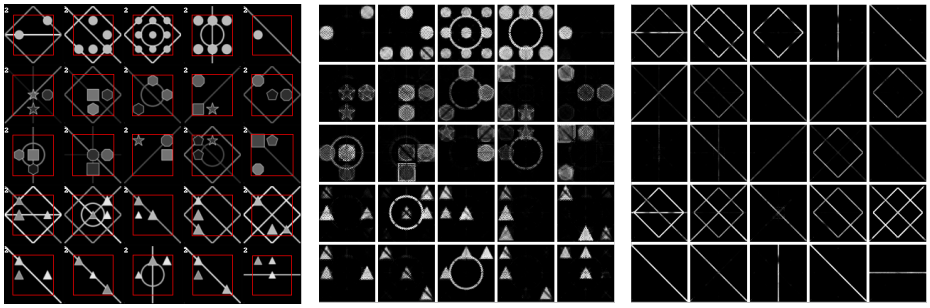

Fig. 6. AIR decomposes PGM frames (left) into grid and background slots (centre, right). Red bounding boxes denote attention windows for the first slot.

Table 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 [30].

PGM set	Wild-ResNet [20]	WReN	CoPINet [29]	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**

5.3 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 [29], LEN and LEN+T [30]. We detail the results in Table 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 [30] by 15.8ppt.

Table 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.

Method	Acc	Centre	2x2	3x3	L-R	U-D	O-IC	O-IG
WReN [29]	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 [30]	72.9	80.2	57.5	62.1	73.5	81.2	84.4	71.5
LEN+T [30]	78.3	82.3	58.5	64.3	87.0	85.5	88.9	81.9
Human [28]	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. Accuracy (%) of models over RAVEN, given various training set sizes. Accuracy is averaged over all problem configurations.

% 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**

Performance vs. training set size. As in [29], we also explore model performance as a function of training set size, in order to further evaluate the efficiency of our methods. Table 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 [28] 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 4 and 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.

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

Table 4. Generalisation test between `Left-Right` and `Up-Down` configurations. Rows and columns indicate training and test sets respectively.

	`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 5. Generalisation test between `2x2Grid` and `3x3Grid` 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**

`Left-Right` and `Up-Down` configurations – and generalisation between them – falls to around 43% ± 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).

6 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 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 [4, 26, 22]; we expect that such improvements will minimise the need to fine-tune AIR between configurations.

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

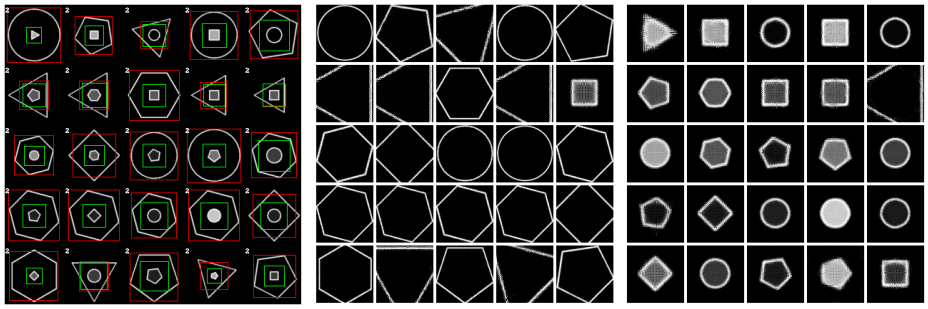

Fig. 7. Visualisation of AIR's decomposition of `Out-InCentre` frames (left) into two slots (centre, right). Bounding boxes denote attention windows.

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 [2, 8], 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. [27]) to be encoded via graph neural networks [21] represents a possible direction in handling the variable length outputs of AIR without padding them.

7 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 paper 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.

References

1. Bingham, E., Chen, J.P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., Goodman, N.D.: Pyro: Deep Universal Probabilistic Programming. Journal of Machine Learning Research (2018)
2. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)
3. Chen, T.Q., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. In: Advances in Neural Information Processing Systems. pp. 2610–2620 (2018)
4. Crawford, E., Pineau, J.: Spatially invariant unsupervised object detection with convolutional neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 3412–3420 (2019)
5. Engelcke, M., Kosiorek, A.R., Jones, O.P., Posner, I.: Genesis: Generative scene inference and sampling with object-centric latent representations. arXiv preprint arXiv:1907.13052 (2019)
6. Eslami, S.A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Hinton, G.E., et al.: Attend, infer, repeat: Fast scene understanding with generative models. In: Advances in Neural Information Processing Systems. pp. 3225–3233 (2016)
7. Gentner, D., Markman, A.B.: Structure mapping in analogy and similarity. American psychologist **52**(1), 45 (1997)
8. Greff, K., Kaufmann, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. arXiv preprint arXiv:1903.00450 (2019)
9. Hahne, L., Lüddecke, T., Wörgötter, F., Kappel, D.: Attention on abstract visual reasoning. arXiv preprint arXiv:1911.05990 (2019)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., Mohamed, S., Lerchner, A.: Early visual concept learning with unsupervised deep learning. arXiv preprint arXiv:1606.05579 (2016)
12. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. Iclr **2**(5), 6 (2017)
13. Hofstadter, D.R.: Analogy as the core of cognition. The analogical mind: Perspectives from cognitive science pp. 499–538 (2001)
14. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
15. Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning. pp. 2649–2658 (2018)
16. Lovett, A., Forbus, K.: Modeling visual problem solving as analogical reasoning. Psychological review **124**(1), 60 (2017)
17. McCarthy, J., Minsky, M., Rochester, N., Shannon, C.: A proposal for the dartmouth summer research project on artificial intelligence (1955). Reprinted online at http://www-formal. stanford. edu/jmc/history/dartmouth/dartmouth. html (2018)
18. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. In: NIPS-W (2017)

19. Raven, J.: The raven's progressive matrices: change and stability over culture and time. Cognitive psychology **41**(1), 1–48 (2000)
20. Santoro, A., Hill, F., Barrett, D., Morcos, A., Lillicrap, T.: Measuring abstract reasoning in neural networks. In: International Conference on Machine Learning. pp. 4477–4486 (2018)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. pp. 593–607. Springer (2018)
22. Stanić, A., Schmidhuber, J.: R-sqair: Relational sequential attend, infer, repeat. arXiv preprint arXiv:1910.05231 (2019)
23. Steenbrugge, X., Leroux, S., Verbelen, T., Dhoedt, B.: Improving generalization for abstract reasoning tasks using disentangled feature representations. arXiv preprint arXiv:1811.04784 (2018)
24. van Steenkiste, S., Locatello, F., Schmidhuber, J., Bachem, O.: Are disentangled representations helpful for abstract visual reasoning? In: Advances in Neural Information Processing Systems. pp. 14222–14235 (2019)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
26. Wang, D., Jamnik, M., Lio, P.: Unsupervised and interpretable scene discovery with discrete-attend-infer-repeat. arXiv preprint arXiv:1903.06581 (2019)
27. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 670–685 (2018)
28. Zhang, C., Gao, F., Jia, B., Zhu, Y., Zhu, S.C.: Raven: A dataset for relational and analogical visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5317–5327 (2019)
29. Zhang, C., Jia, B., Gao, F., Zhu, Y., Lu, H., Zhu, S.C.: Learning perceptual inference by contrasting. In: Advances in Neural Information Processing Systems. pp. 1073–1085 (2019)
30. Zheng, K., Zha, Z.J., Wei, W.: Abstract reasoning with distracting features. In: Advances in Neural Information Processing Systems. pp. 5834–5845 (2019)
31. Zhuo, T., Kankanhalli, M.: Solving raven's progressive matrices with neural networks. arXiv preprint arXiv:2002.01646 (2020)