LEARNING TO REASON OVER VISUAL OBJECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

A core component of human intelligence is the ability to identify abstract patterns governing complex, high-dimensional perceptual data, as exemplified by visual reasoning tasks such as Raven's Progressive Matrices (RPM). Motivated by the goal of designing AI systems with this capacity, recent work has focused on evaluating whether neural networks can learn to solve RPM-like problems. This work has generally found that strong performance on these problems requires the incorporation of inductive biases that are specific to the RPM problem format, raising the question of whether such models might be more broadly useful. Here, we investigated the extent to which a general-purpose mechanism for processing visual scenes in terms of objects might enable abstract visual reasoning. We found that a simple model, consisting only of an object-centric encoder and a transformer reasoning module, displayed performance approaching state-of-the-art methods on two challenging RPM-like benchmarks (PGM and I-RAVEN), suggesting that an inductive bias for object-centric processing is a key component of visual reasoning, and may supplant some problem-specific inductive biases.

1 INTRODUCTION

Human reasoning is driven by a capacity to extract simple, low-dimensional abstractions from complex, high-dimensional inputs. We perceive the world around us in terms of objects, relations, and higher order patterns, allowing us to generalize beyond the sensory details of our experiences, and make powerful inferences about novel situations. This capacity for abstraction is particularly well captured by visual analogy problems, in which the reasoner must abstract over the superficial details of visual inputs, in order to identify a common higher order pattern. An especially challenging example of these kinds of problems are the Raven's Progressive Matrices (RPM) problem sets (Raven, 1938), which have been found to be especially diagnostic of human reasoning abilities (Snow et al., 1984).

A growing body of recent work has aimed to build learning algorithms that capture this capacity for abstract visual reasoning. Much of this work has revolved around two recently developed benchmarks – the Procedurally Generated Matrices (PGM) (Barrett et al., 2018), and the RAVEN dataset (Zhang et al., 2019a) – consisting of a large number of automatically generated RPM-like problems. As in RPM, each problem consists of a 3×3 matrix populated with geometric forms, in which the bottom right cell is left blank. The challenge is to infer the abstract pattern that governs the matrix, and use that inferred pattern to 'fill in the blank', by selecting from a set of choices. As can be seen in Figure 1, these problems can be quite complex, with potentially many objects per cell, and multiple rules per problem, yielding a highly challenging visual reasoning task.

There is substantial evidence that human visual reasoning is fundamentally organized around the decomposition of visual scenes into objects (Duncan, 1984; Pylyshyn, 1989; Peters & Kriegeskorte, 2021). Objects offer a simple, yet powerful, low-dimensional abstraction that captures the inherent compositionality underlying visual scenes. Despite the centrality of objects in visual reasoning, previous works have so far not explored the use of object-centric representations in abstract visual reasoning tasks such as RAVEN and PGM, or at best have employed an imprecise approximation to object representations based on spatial location.

Recently, a number of methods have been proposed that enable the extraction of precise object-centric representations directly from pixel-level inputs, without the need for veridical segmentation data (Greff et al., 2019; Burgess et al., 2019; Locatello et al., 2020; Engelcke et al., 2021). Here,

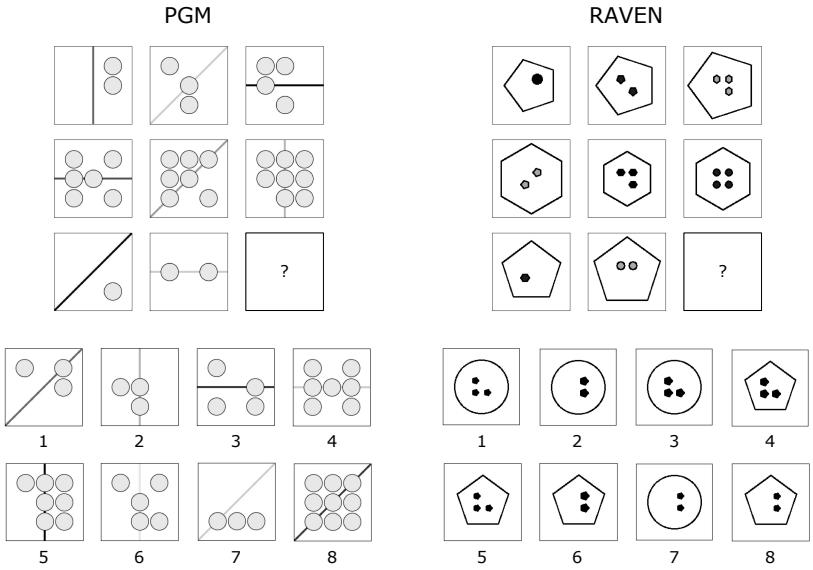

Figure 1: Example problems from the Procedurally Generated Matrices (PGM) (Barrett et al., 2018) and RAVEN (Zhang et al., 2019b) datasets. Problem structure is based on Raven's Progressive Matrices (RPM) (Raven, 1938). Each problem consists of a 3×3 matrix, with geometric forms governed by a set of abstract rules. The reasoner must infer these rules, and use them to select the best of the available choices to complete the problem.

we leveraged these recent innovations, and developed a model that combines an object-centric encoding method, slot attention (Locatello et al., 2020), with a generic transformer-based reasoning module (Vaswani et al., 2017). The combined system, which we term the *Slot Transformer Scoring Network* (STSN, Figure 2) achieves performance approaching that of state-of-the-art methods on both PGM and I-RAVEN (a more challenging variant of RAVEN), despite its general-purpose architecture. These results suggest that object-centric encoding is an essential component for achieving strong abstract visual reasoning, and may obviate the need for some problem-specific inductive biases.

2 Related Work

Since the introduction of the PGM (Barrett et al., 2018) and RAVEN (Zhang et al., 2019a) datasets, a number of methods have been proposed for learning to solve RPM-like problems Barrett et al. (2018); Steenbrugge et al. (2018); Zhang et al. (2019b); Zheng et al. (2019); Spratley et al. (2020); Jahrens & Martinetz (2020); Wang et al. (2020); Wu et al. (2020); Benny et al. (2021); Hu et al. (2021); Zhuo & Kankanhalli (2022). Though significant progress has been made, the best performing methods generally rely on inductive biases that are specifically tailored to the format of these datasets. For instance, the Scattering Compositional Learner (SCL) (Wu et al., 2020), arguably the best current model (achieving strong performance on both PGM and RAVEN), assumes that rules will be independently applied in each feature dimension, with no interaction between features. Similarly, the Multi-Scale Relation Network (MRNet) (Benny et al., 2021), which achieves strong performance on PGM, explicitly builds the row-wise and column-wise structure of RPM problems into its architecture. These approaches therefore raise the question of whether such problem-specific inductive biases are necessary to achieve strong performance on these problems.

In this work, we explore the utility of a more general-purpose inductive bias – a mechanism for processing visual scenes in terms of objects. In contrast, most previous approaches to solving RPM-like problems have operated over embeddings of entire image panels, and thus likely fail to capture the the compositional structure of such multi-object visual inputs. A few works have attempted to approximate object-centric representations, by, for instance, treating spatial location as a proxy for objects (Wu et al., 2020), or by employing encodings at different spatial scales (Benny et al.,

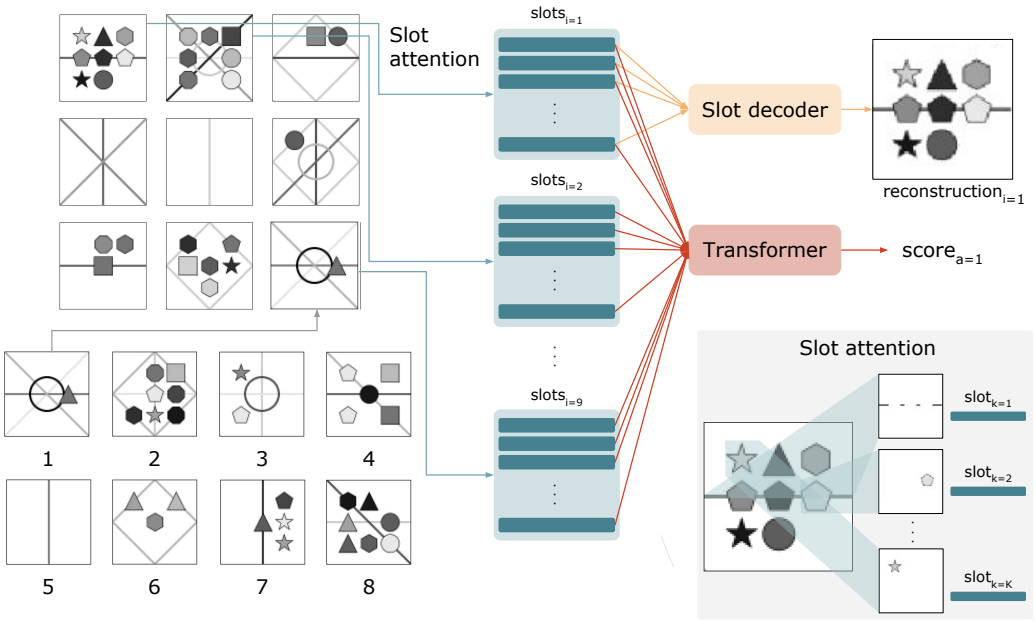

Figure 2: Slot Transformer Scoring Network (STSN). STSN combines slot attention, an object-centric encoding method, and a transformer reasoning module. Slot attention decomposes each image panel into a set of K slots, which are randomly initialized and iteratively updated through competitive attention over the image. STSN assigns a score to each of the 8 potential answers, by independently evaluating the combination of each answer choice together with the 8 context panels. For each answer choice, slots are extracted from that choice, and the context panels, and these slots are concatenated to form a sequence which is passed to the transformer, which generates a score. The scores for all answer choices are passed through a softmax in order to compute the task loss $\mathcal{L}_{task}$. Additionally, the slots for each image panel are passed through a slot decoder, yielding a reconstruction of that image panel, from which the reconstruction loss $\mathcal{L}_{recon}$ is computed.

2021) (therefore preferentially capturing larger vs. smaller objects), but these approximations likely failed to extract precise object-centric representations, especially in problems with many overlapping objects, such as PGM.

Recently, a number of methods have been proposed to address the challenging task of annotation-free object segmentation (Greff et al., 2019; Burgess et al., 2019; Locatello et al., 2020; Engelcke et al., 2021). In this approach, the decomposition of a visual scene into objects is treated as a latent variable to be inferred in the service a downstream objective, such as autoencoding, without access to any veridical segmentation data. Here, we utilized the slot attention method (Locatello et al., 2020), but we expect that our approach should be compatible with other object-centric encoding methods as well.

Our method employs a generic transformer (Vaswani et al., 2017) to perform reasoning over the object-centric representations extracted by slot attention. This approach allows the natural permutation invariance of objects to be preserved in the reasoning process. A few other recent works have employed systems that provide object-centric representations as the input to a transformer network (Ding et al., 2021; Wu et al.), most notably ALOE (Attention over Learned Object Embeddings (Ding et al., 2021)), which employed a different object encoding method (MONet (Burgess et al., 2019)). These works have shown strong visual reasoning performance for such systems in other task domains, including question answering from video, underscoring the broad applicability of this general approach.

3 APPROACH

3.1 PROBLEM DEFINITION

Each RPM problem consists of a 3×3 matrix of panels where each row (or each column) is explained by a common set of abstract rules. The last panel, the entry in the third column of the third row, is missing and must be filled from a set of eight candidate panels so as to best complete the matrix in accordance with the abstract rules. Each panel is an image consisting of different number of objects with attributes like size, shape and color. Formally, each RPM problem consists of 16 image panels $X = \{x_i\}_{i=1}^{16}$, where the first 8 image panels are context images $X_c = \{x_i\}_{i=1}^{8}$, and the last 8 image panels are candidate answer images $X_a = \{x_i\}_{i=9}^{16}$. The context images X_c are placed in the 3×3 matrix with the last entry blank. The task is to fill the blank entry with the correct answer $y \in X_a$.

3.2 OBJECT-CENTRIC ENCODER

STSN employs slot attention (Locatello et al., 2020) to extract object-centric representations. Slot attention first performs some initial processing of the images using a convolutional encoder. Then, a set of K slots compete to attend over the locations in the feature map produced by this encoder. We set the value of K to be equal to the maximum number of objects possible in a given image panel. For each image, slots are randomly initialized from a distribution with shared mean and variance, and then iteratively updated based on a transformer-style attention operation. Specifically, each slot emits a query through a linear projection, and each location in the feature map emits a key and value. A dot product query-key attention followed by softmax is then used to generate the attention weights. The weighted mean of the values are used to update the slot representations using a Gated Recurrent Unit (Cho et al., 2014) followed by a residual MLP with ReLU activations. More details can be found in Locatello et al. (2020). After T iterations of slot attention, we pass the resulting slots through a reasoning module, which we describe next.

3.3 REASONING MODULE

After object representations are extracted by slot attention, they are then passed to a transformer (Vaswani et al., 2017). For each candidate answer choice, the transformer operates over the slots obtained from the 8 context images, and the image for that answer choice. We flattened the slots over the dimensions representing the number of slots and images, such that, for each cnadidate answer, there were $9K$ slot vectors of dimension D. We then applied Temporal Context Normalization (TCN) (Webb et al., 2020a), which has been shown to significantly improve out-of-distribution in relational tasks. To give the model knowledge about which slot representation corresponded to which row and column of the matrix, we added a learnable linear projection from one-hot encodings of the row and column indices (after applying TCN). We concatenated a learned CLS token (analogous to CLS token in Devlin et al. (2018)) of dimension D, before passing it through a transformer with L layers and H self attention heads. The transformed value of the CLS token was passed through a linear output unit to generate a score for each candidate answer image.

3.4 OPTIMIZATION

The entire model was trained end-to-end (except when training the model on the extrapolation regime of PGM, as detailed in Section 4.3) to optimize two objectives. First, we computed a reconstruction loss $\mathcal{L}_{recon}$, the mean squared error between the 16 image panels and their reconstructed outputs, in order to encourage the model to utilize slot attention in an object-centric manner. To generate the reconstructed outputs, we passed the slot representations through the slot decoder used in Locatello et al. (2020). Second, we computed a task loss $\mathcal{L}_{task}$, the cross entropy loss between the target answer and the scores for each of the candidate answers. These two losses were combined to form the final loss $\mathcal{L} = \lambda * \mathcal{L}_{recon} + \mathcal{L}_{task}$, where λ is a hyperparameter that controls the relative strength of the reconstruction loss.

4 EXPERIMENTS

4.1 DATASETS

PGM. The PGM dataset was introduced by Barrett et al. (2018), and consists of problems belonging to eight different regimes with different generalization difficulty. Each matrix problem in PGM is defined by the abstract structure, $\mathcal{S} = \{[r, o, a] : r \in \mathcal{R}, o \in \mathcal{O}, a \in \mathcal{A}\}$, where $\mathcal{R}$ ={progression, XOR, AND, OR, consistent union} are the set of rules, $\mathcal{O}$ ={shape, line} are the set of objects, and $\mathcal{A}$ ={size, type, position, color, number} are the set of attributes. Each regime consists of 1.2M training problems, 20K validation problems, and 200K testing problems. Due to the enormous size of the dataset we focused on the neutral and extrapolation regimes. More details can be found in Barrett et al. (2018).

I-RAVEN. The RAVEN dataset was introduced by Zhang et al. (2019a), with problems belonging to seven different configurations. These configurations are defined by the spatial layout of the elements in each panel, ranging from low visual complexity (e.g., the 'Center' configuration, in which each panel contains just a single object in the center of the image), to high visual complexity (e.g., the 'O-IG' regime, in which each panel contains an outer object surrounding an inner grid of objects). Some configurations have multiple components $\mathcal{C}$ to which separate rules can be bound. Thus, each problem in RAVEN is defined by the abstract structure $\mathcal{S} = \{[r, c, a] : r \in \mathcal{R}, c \in \mathcal{C}, a \in \mathcal{A}\}$, where $\mathcal{R}$ ={constant, progression, arithmetic, distribute three} are the set of rules, $\mathcal{C}$ are the set of components (depending on the particular configuration), and $\mathcal{A}$ ={number, position, size, type, color} are the set of attributes. There are a total of 42K training problems, 14K validation problems, and 14K testing problems. We trained STSN jointly on all configurations in RAVEN.

It was subsequently discovered that the original RAVEN dataset employed a biased method for generating candidate answers, that could be exploited so as to achieve near perfect performance by only viewing these candidate answers (i.e., ignoring the problem itself) (Hu et al., 2021). To address this, Hu et al. (2021) proposed the Impartial RAVEN (I-RAVEN) dataset, with an unbiased procedure for generating candidate answers. As with most recent work in this domain, we performed our evaluation on I-RAVEN.

4.2 BASELINES

We compared our model to several baselines, as detailed in Tables 1 and 2. To the best of our knowledge, these baselines include the current best performing models on the I-RAVEN and PGM benchmarks. We didn't use any auxiliary information (i.e., training to explicitly label the underlying rules), and hence for fair comparison we only compared to baselines that didn't use auxiliary loss.

4.3 EXPERIMENTAL DETAILS

The convolutional encoder used to extract feature maps had similar hyperparameters as in Locatello et al. (2020), consisting of 4 convolutional layers, each with a kernel size of 5, 32 channels, and padding of 2. Following this, positional embeddings were added, and feature maps were flattened along the spatial dimension, followed by layer normalization (Ba et al., 2016), and two position-wise linear layers with 32 units and ReLU non-linearities. We resized the images to a resolution of 80×80, and normalized the pixels values to be in the range [0,1]. We also applied random online augmentations during training, including horiztonal and vertical flips, rotations by multiples of $90°$, and brightness changes by a factor in the range [0.5,1.5]. Note we applied the same augmentations to all the image panels in a given problem, so that the abstract rule remained the same. For I-RAVEN, we set the value of slots per image to $K = 9$, and for PGM to $K = 16$. The number of slot attention iterations was set to $T = 3$, and the dimensionality of the slots was set to $D = 32$. The decoder consisted of 4 transposed convolutional layers, with kernel sizes $\{5, 5, 5, 3\}$, channel numbers of $\{32, 32, 32, 2\}$, and padding sizes of $\{2, 2, 2, 1\}$.

We used a transformer with $L = 6$ layers, $H = 8$ attention heads, and a feedforward dimension of size 512. On the neutral regime of the PGM dataset we used a transformer with $L = 24$ layers. We found that $\lambda = 1000$ provided sufficient weight on the reconstruction objective, which encouraged the model to use object-centric representations. For I-RAVEN, we trained with a batch size of 16 (4 V100 GPUs with 16GB of RAM, or 1 A100 GPU with 40GB of RAM) for a maximum of 300

Table 1: Results on I-RAVEN.

Model	Test Accuracy (%)							
	Average	Center	2Grid	3Grid	L-R	U-D	O-IC	O-IG
LSTM (Hu et al., 2021)	18.9	26.2	16.7	15.1	14.6	16.5	21.9	21.1
WReN Hu et al. (2021)	23.8	29.4	26.8	23.5	21.9	21.4	22.5	21.5
ResNet (Hu et al., 2021)	40.3	44.7	29.3	27.9	51.2	47.4	46.2	35.8
Wild ResNet (Hu et al., 2021)	44.3	50.9	33.1	30.8	53.1	52.6	50.9	38.7
LEN (Zheng et al., 2019)	39.0	45.5	27.9	26.6	44.2	43.6	50.5	34.9
CoPINet (Zhang et al., 2019b)	46.3	54.4	33.4	30.1	56.8	55.6	54.3	39.0
SRAN (Hu et al., 2021)	63.9	80.1	53.3	46.0	72.8	74.5	71.0	49.6
SCL (Wu et al., 2020)	95.0	99.0	96.2	89.5	97.9	97.1	97.6	87.7
STSN (ours)	93.8	98.0	93.0	85.0	96.5	96.9	96.6	90.4

epochs (the epoch with the highest validation accuracy was selected for evaluation on the test set) with a learning rate of 0.0004. We used learning rate warmup (Goyal et al., 2017) for 75K iterations. For PGM, we trained with a batch size of 96 (6 A100 GPUs) for 43 epochs on the neutral regime, and 50 epochs on the extrapolation regime with a learning rate of 0.00008. We used a learning rate warmup for 10K iterations. We used the ADAM optimizer (Kingma & Ba, 2014) and all experiments were performed using the Pytorch library (Paszke et al., 2017).

For I-RAVEN, we report results from the best out of 3 trained models. For comparison, some previous works report results from the best out of 5 trained models (Wu et al., 2020). Due the random initialization of the slots for each problem, there is a small amount of random variation in the performance of the same trained model when evaluated multiple times on the test set. We evaluated the best trained model 5 times and selected the best result. The mean and standard error of these evaluations is reported in the Appendix (Table 4). For PGM, we only trained 1 model on the neutral regime, and 1 model on the extrapolation regime, due to the computational cost of training models on such a large dataset. We also only evaluated these models once on their respective test sets.

For PGM, we pretrained the convolutional encoder, slot attention, and slot decoder on the reconstruction objective. For the PGM neutral regime, we pretrained on the neutral training set, and fine-tuned while training on the primary task. For the PGM extrapolation regime, we also pretrained (only the convolutional encoder, slot attention, and slot decoder) on the neutral training set, and then froze the parameters, to give slot attention exposure to a broader range of feature values. However, training of the reasoning module on the PGM task was restricted to the extrapolation training set.

4.4 RESULTS

Table 1 shows the results on the I-RAVEN dataset. The results show the test accuracy for each of the configurations, along with the average across configurations. Overall, STSN performed very close (93.8% accuracy) to the current state-of-the-art, SCL (95.0% accuracy). STSN also achieved a new state-of-the-art (90.4% accuracy) on the most difficult configuration, 'O-IG', consisting of a large object surrounding an inner grid of smaller objects, perhaps due to an enhanced ability to parse individual objects.

Table 2 shows the results on the PGM dataset. On the neutral regime, STSN performed comparably (84.6% accuracy) to other strong models, including Rel-Base (85.5% accuracy), and SCL (88.9% accuracy). MRNet performs better (93.4% accuracy) than these models (and better than STSN) on PGM, but has not been evaluated on I-RAVEN. MLRN stands out on the PGM dataset with a nearly perfect level of performance (98.0% accuracy). However, Zhuo & Kankanhalli (2022) evaluated MLRN on RAVEN and found that it performed at chance level (12.5% accuracy), suggesting that some aspect of its architecture may be overfit to the PGM dataset. Overall, STSN's performance approaches the current best performing model on both the I-RAVEN and PGM datasets (SCL), despite its relatively general-purpose architecture. It is worth noting that the training and validation timecourses for STSN suggest it may not have fully converged (see Figure 7 in the Appendix), so it may show further improvement with additional training.

Table 2: Results on PGM.

Model	Test Accuracy (%)	
	Neutral	Extrapolation
CNN+MLP (Barrett et al., 2018)	33.0	-
CNN+LSTM (Barrett et al., 2018)	35.8	-
ResNet-50 (Barrett et al., 2018)	42.0	-
Wild-ResNet (Barrett et al., 2018)	48.0	-
CoPINet (Zhang et al., 2019b)	56.4	-
WReN ($\beta = 0$) (Barrett et al., 2018)	62.6	17.2
VAE-WReN (Steenbrugge et al., 2018)	64.2	-
MXGNet ($\beta = 0$) (Wang et al., 2020)	66.7	18.9
LEN ($\beta = 0$) (Zheng et al., 2019)	68.1	-
DCNet (Zhuo & Kankanhalli, 2022)	68.6	17.8
T-LEN ($\beta = 0$) (Zheng et al., 2019)	70.3	-
SRAN (Hu et al., 2021)	71.3	-
Rel-Base (Spratley et al., 2020)	85.5	22.1
SCL (Wu et al., 2020)	88.9	-
MRNet (Benny et al., 2021)	93.4	19.2
MLRN (Jahrens & Martinetz, 2020)	98.0	14.9
STSN (ours)	84.6	17.3

Table 3: Ablation study on the I-RAVEN dataset.

Model	Test Accuracy (%)							
	Average	Center	2Grid	3Grid	L-R	U-D	O-IC	O-IG
STSN	93.8	98.0	93.0	85.0	96.5	96.9	96.6	90.4
STSN (w/o slot attention)	71.0	90.0	71.0	59.4	73.8	75.4	74.5	53.0
STSN (w/o TCN)	86.5	97.0	76.0	69.2	96.0	96.0	95.6	75.8
STSN (w/o augmentations)	90.3	96.1	88.3	80.8	93.2	93.7	94.8	85.2

On the extrapolation regime of the PGM dataset, all current models perform poorly, including STSN (17.3% accuracy), suggesting the need for further development on this challenging out-of-distribution regime.

4.5 Ablation Study

We analysed the importance of the different components of STSN through some ablation studies on the I-RAVEN dataset (Table 3). First, we removed the slot attention module from STSN, by averaging the value embeddings from the input feature vectors over the image space (i.e. using only a single slot per panel). The average test accuracy decreased by more than 20%, suggesting that object-centric representations play a critical role in the model's performance. Second, we studied the performed an ablation on TCN, resulting in a test accuracy decrease of around 7%, in line with previous findings demonstrating a role of TCN in improved generalization (Webb et al., 2020a). Finally, we performed an ablation on the image augmentations performed during training, resulting in a test accuracy decrease of more than 3%, suggesting that the augmentations also helped to improve generalization. Overall, these results show that the use of object-centric representations was the most important factor explaining STSN's performance on this task.

4.6 Visualization of object masks

We also visually inspected the attention behavior of STSN's slot attention module (Figure 3). We found that STSN's slot-specific reconstructions conformed nearly perfectly to the individual objects in the image panels of PGM, with the remaining slots left unused. This confirms that STSN was engaged in object-centric processing. We also evaluated STSN on I-RAVEN with a range of values for λ, and found that with lower values of λ, STSN's reconstructions were no longer object-centric,

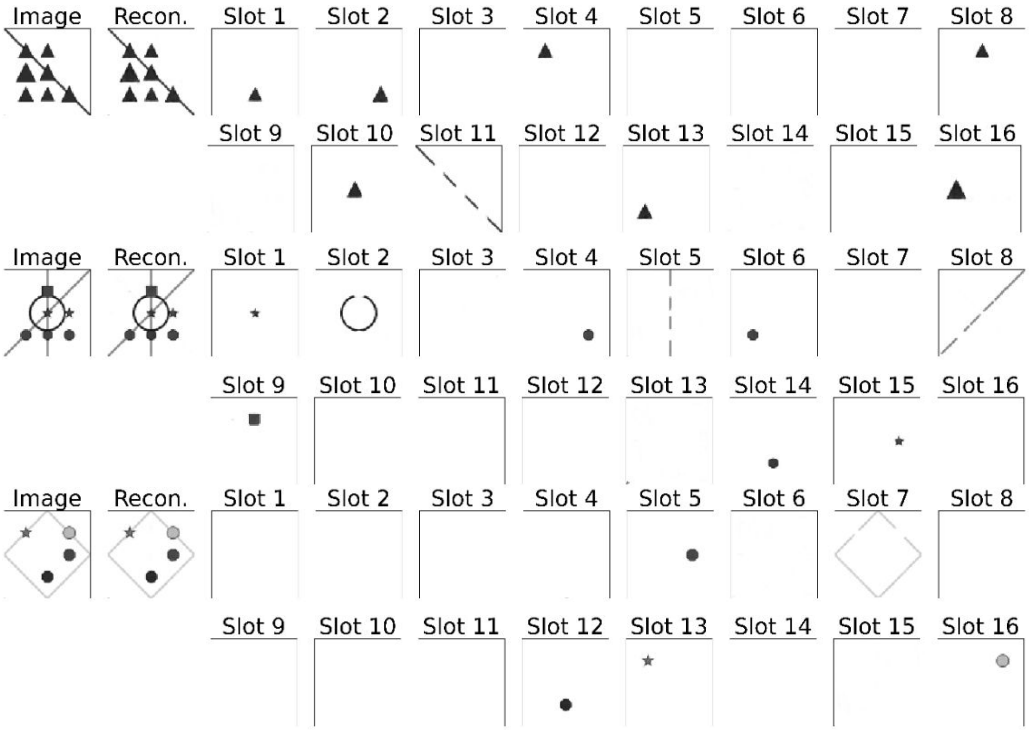

Figure 3: Slot-specific reconstructions generated by STSN. 3 problems were chosen at random from the PGM neutral test set. The first two images for each problem show the original image and the combined reconstruction. The next 16 images show the slot-specific reconstruction for each of the 16 slots. In general, STSN's slot attention module implemented a nearly perfect object-based segmentation of its input images, despite receiving no veridical segmentation information during training or test.

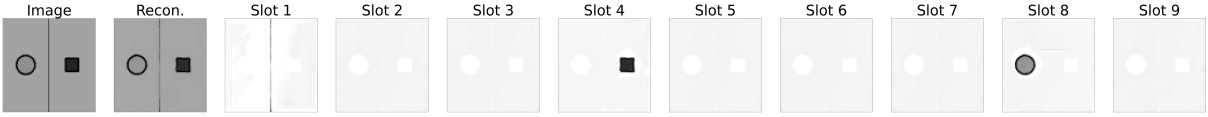

Figure 4: Slot-specific reconstructions generated by STSN for $\lambda = 1000$ on I-RAVEN.

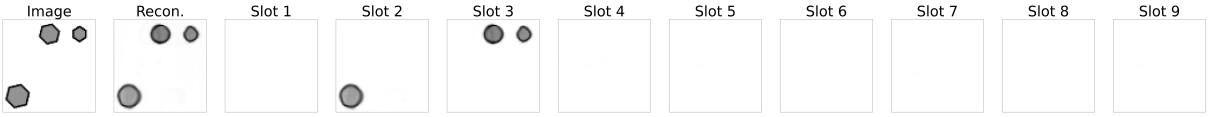

Figure 5: Slot-specific reconstructions generated by STSN for $\lambda = 100$ on I-RAVEN.

Figure 6: Slot-specific reconstructions generated by STSN for $\lambda = 1$ on I-RAVEN.

suggesting that a sufficiently high weight on this reconstruction objective was important for encouraging object-centric representations (Figures 4-6).

5 Conclusion and Future Directions

We have presented a simple, general-purpose visual reasoning model, organized around the principle of object-centric processing. On two challenging visual reasoning benchmarks, PGM and I-RAVEN, we found that our proposed model (STSN) displayed performance approaching state-of-the-art models, despite the relative lack of problem-specific inductive biases. These results suggest that object-centric processing is a powerful inductive bias for visual reasoning problems such as RPM.

Some previous work has proposed novel relational inductive biases for the purposes of achieving strong out-of-distribution generalization in visual reasoning problems Webb et al. (2020b); Zhang et al. (2021); Kerg et al. (2022). This work has often assumed (i.e. hand-coded) object-centric representations. We view our approach as complementary with these previous approaches, and suggest that a fruitful avenue for future work will be to pursue the integration of object-centric and relational inductive biases.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. Measuring abstract reasoning in neural networks. In *International conference on machine learning*, pp. 511–520. PMLR, 2018.

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

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

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

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

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

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34:8085–8094, 2021.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pp. 2424–2433. PMLR, 2019.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1567–1574, 2021.

Marius Jahrens and Thomas Martinetz. Solving raven's progressive matrices with multi-layer relation networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2020.

Giancarlo Kerg, Sarthak Mittal, David Rolnick, Yoshua Bengio, Blake Richards, and Guillaume Lajoie. On neural architecture inductive biases for relational tasks. *arXiv preprint arXiv:2206.05056*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Benjamin Peters and Nikolaus Kriegeskorte. Capturing the objects of vision with neural networks. *Nature Human Behaviour*, 5(9):1127–1144, 2021.

Zenon Pylyshyn. The role of location indexes in spatial perception: A sketch of the finst spatial-index model. *Cognition*, 32(1):65–97, 1989.

John C Raven. *Progressive matrices: A perceptual test of intelligence, individual form*. London: Lewis, 1938.

Richard E Snow, Patrick C Kyllonen, Brachia Marshalek, et al. The topography of ability and learning correlations. *Advances in the psychology of human intelligence*, 2(S 47):103, 1984.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Duo Wang, Mateja Jamnik, and Pietro Lio. Abstract diagrammatic reasoning with multiplex graph networks. *arXiv preprint arXiv:2006.11197*, 2020.

Taylor Webb, Zachary Dulberg, Steven Frankland, Alexander Petrov, Randall O'Reilly, and Jonathan Cohen. Learning representations that support extrapolation. In *International conference on machine learning*, pp. 10136–10146. PMLR, 2020a.

Taylor W Webb, Ishan Sinha, and Jonathan D Cohen. Emergent symbols through binding in external memory. *arXiv preprint arXiv:2012.14601*, 2020b.

Yuhuai Wu, Honghua Dong, Roger Grosse, and Jimmy Ba. The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *arXiv preprint arXiv:2007.04212*, 2020.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Jiaqi Xi, Thomas Kipf, and Animesh Garg. Slotformer: Long-term dynamic modeling in object-centric models. In *UAI 2022 Workshop on Causal Representation Learning*.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5317–5327, 2019a.

Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. Learning perceptual inference by contrasting. *Advances in neural information processing systems*, 32, 2019b.

Chi Zhang, Sirui Xie, Baoxiong Jia, Ying Nian Wu, Song-Chun Zhu, and Yixin Zhu. Learning algebraic representation for systematic generalization in abstract reasoning. *arXiv preprint arXiv:2111.12990*, 2021.

Kecheng Zheng, Zheng-Jun Zha, and Wei Wei. Abstract reasoning with distracting features. *Advances in Neural Information Processing Systems*, 32, 2019.

Tao Zhuo and Mohan Kankanhalli. Effective abstract reasoning with dual-contrast network. *arXiv preprint arXiv:2205.13720*, 2022.

A APPENDIX

Table 4: STSN results on I-RAVEN, showing variance across test runs for the same trained model.

	Test Accuracy (%)							
Model	Average	Center	2Grid	3Grid	L-R	U-D	O-IC	O-IG
STSN	$93.4^{\pm 0.11}$	$97.8^{\pm 0.05}$	$92.5^{\pm 0.14}$	$84.7^{\pm 0.12}$	$96.4^{\pm 0.08}$	$96.7^{\pm 0.08}$	$96.5^{\pm 0.10}$	$89.6^{\pm 0.20}$

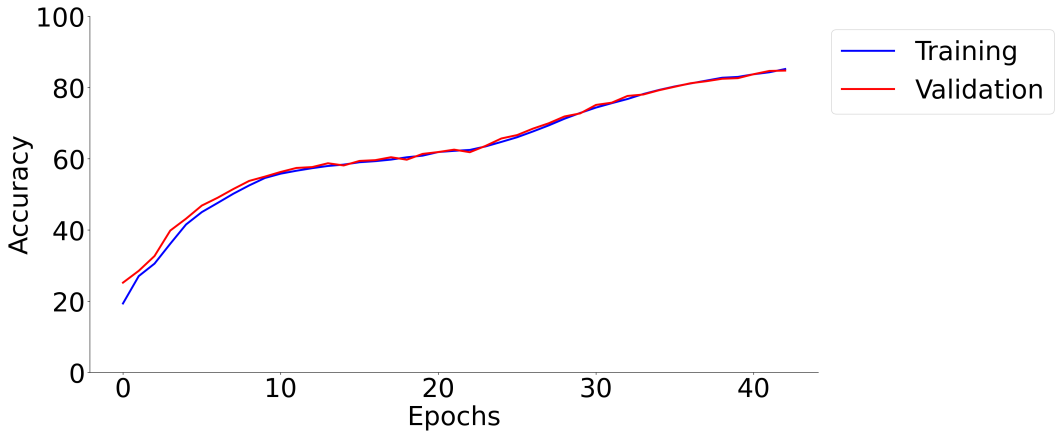

Figure 7: Training and validation timecourses for STSN on the PGM neutral dataset.