

SAViR-T: Spatially Attentive Visual Reasoning with Transformers

Pritish Sahu ✉, Kalliopi Basioti ✉, and Vladimir Pavlovic ✉

Dept. of Computer Science, Rutgers University, NJ, USA
pritch.sahu@rutgers.edu, kib21@scarletmail.rutgers.edu,
vladimir@cs.rutgers.edu

Abstract. We present a novel computational model, SAViR-T, for the family of visual reasoning problems embodied in the Raven’s Progressive Matrices (RPM). Our model considers explicit spatial semantics of visual elements within each image in the puzzle, encoded as spatio-visual tokens, and learns the intra-image as well as the inter-image token dependencies, highly relevant for the visual reasoning task. Token-wise relationship, modeled through a transformer-based SAViR-T architecture, extract group (row or column) driven representations by leveraging the group-rule coherence and use this as the inductive bias to extract the underlying rule representations in the top two row (or column) per token in the RPM. We use this relation representations to locate the correct choice image that completes the last row or column for the RPM. Extensive experiments across both synthetic RPM benchmarks, including RAVEN, I-RAVEN, RAVEN-FAIR, and PGM, and the natural image-based “V-PROM” demonstrate that SAViR-T sets a new state-of-the-art for visual reasoning, exceeding prior models’ performance by a considerable margin.

Keywords: Abstract Visual Reasoning, Raven’s Progressive Matrices, Transformer

1 Introduction

Human abstract reasoning is the analytic process aimed at decision-making or solving a problem [1]. In the realm of visual reasoning, humans find it advantageous, explicitly or implicitly, to break down an image into well-understood low-level concepts before proceeding with the reasoning task, e.g., examining object properties or counting objects. These low-level concepts are combined to form high-level abstract concepts in a list of images, enabling relational reasoning functions such as assessing the increment of the object count, changes in the type of object, or object properties, and subsequently applying the acquired knowledge to unseen scenarios. However, replicating such reasoning processes in machines is particularly challenging [12].

A popular test format of abstract reasoning in the visual domain is the Raven’s Progressive Matrix (RPM), developed on Spearman’s work on human

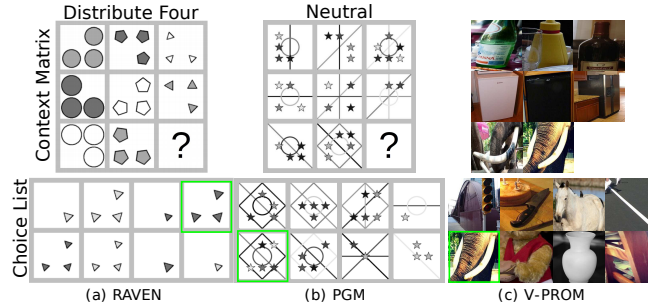


Fig. 1: Three RPM examples from the (a) RAVEN, (b) PGM, and (c) V-PROM datasets. The highlighted green box shows the correct image in the choice list. Solving these RPM requires identifying the underlying rules applied to image attributes or lines along a row or column and, the image with the best fit from the choice list to the rules is the correct answer. RAVEN example (left), the rules governing are “distribute_three” on {position, color } and “progression” on {type, size}. The rules governing PGM(middle) are “OR” on object type line and “XOR” on the position of shapes. And V-PROM(right) example is “And”.

general intelligence [23]. The test is designed as an incomplete 3×3 matrix, with each matrix element being an image and the bottom right location left empty, c.f., Figure 1. Every image can contain one or more objects or lines characterized by the attributes of shape, color, scale, rotation angle, and, holistically, the counts of items or their variability (all circles, all pentagons, both pentagons, and circles, etc.). The top two rows or columns follow a certain unknown rule applied to the attributes; the task is to pick the correct image from an unordered set of choices, satisfying the same constraints. For example in Figure 1, we present three instances of RPM, RAVEN [32], PGM [2], and V-PROM [25] datasets, where the first eight images are denoted as context images, and below them is the set of choice images.

Classical computational models for solving RPMs, built upon access to symbolic attribute representations of the images [5,15,16,17], are incapable of adapting to unseen domains. The success of deep models in other computer vision tasks made it possible to exploit the feature representation and relational learning concepts in visual reasoning [18]. Initial studies [32,2] using widely popular neural network architecture such as ResNet [9] and LSTM [10] failed in solving general reasoning tasks. These models aim to discover underlying rules by mapping the eight context images to each choice image. Modeling the reasoning network [11,33,3] to mimic the human reasoning process has led to a large performance gain. All recent works utilize an encoding mechanism to extract the features/attributes of single or groups of images, followed by a reasoning model that learns the underlying rule from the extracted features to predict scores for images in the choice list. The context features are contrasted against each choice features to elucidate the best image in the missing location. However, these mod-

els make use of holistic image representations, which ignore the important local, spatially contextualized features. Because typical reasoning patterns use intra and inter-image object-level relationships, holistic representations are likely to lead to suboptimal reasoning performance of the models that rely on it.

In this work, we focus on using local, spatially-contextualized features accompanied by an attention mechanism to learn the rule constraint within and across groups (rows or columns). We use a bottom-up and top-down approach to the visual encoding and the reasoning process. From the bottom-up, we address how a set of image regions are associated with each other via the self-attention mechanism from visual transformers. Specifically, instead of extracting a traditional holistic feature vector on image-level, we constrain semantic visual tokens to attend to different image patches. The top-down process is driven to solve visual reasoning tasks that predict an attention distribution over the image regions. To this end, we propose “SAViR-T”, Spatially Attentive Visual Reasoning with Transformers, that naturally integrates the attended region vectors with abstract reasoning. Our reasoning network focuses on entities of interest obtained from the attended vector since the irrelevant local areas have been filtered out. Next, the reasoning task determines the Principal Shared rules in the two complete groups (typically, the top two rows) of the RPM per local region, which are then fused to provide an integrated rule representation. We define a similarity metric to compare the extracted rule representation with the rules formed in the last row when placing each choice at the missing location. The choice with the highest score is predicted as the correct answer. Our contributions in this work are three-fold:

- We propose a novel abstract visual reasoning model, SAViR-T, using spatially-localized attended features for reasoning tasks. SAViR-T accomplishes this using: the Backbone Network responsible for extracting a set of image region encodings; the Visual Transformer performing self-attention on the tokenized feature maps; and finally Reasoning Network, which elucidates the rules governing the puzzle over rows-columns to predict the solution to the RPM.
- SAViR-T automatically learns to focus on different semantic regions of the input images, addressing the problem of extracting holistic feature vectors per image, which may omit critical objects at finer visual scales. Our approach is generic because it is suitable for any configuration of the RPM problems without the need to modify the model for different image structures.
- We drastically improved the reasoning accuracy over all RPM benchmarks, echoed in substantial enhancement on the “ 3×3 Grid”, “Out-In Single”, and “Out Single, In Four Distribute” configurations for RAVEN and I-RAVEN, with strong accuracy gains in the other configurations. We show an average improvement of 2 – 3% for RAVEN-type and PGM datasets. Performance improvement of SAViR-T on V-PROM, a natural image RPM benchmark, significantly improves by 10% over the current state-of-the-art models.

2 Related Works

2.1 Abstract Visual Reasoning

RPM is a form of non-verbal assessment for human intelligence with strong roots in cognitive science [5,2,32]. It measures an individual’s eductive ability, i.e., the ability to find patterns in the apparent chaos of a set of visual scenes [23]. RPM consists of a context matrix of 3×3 that has eight images and a missing image at the last row last column. The participant has to locate the correct image from a choice set of size eight. In the early stages, RPM datasets were created manually, and the popular computational models solved them using hand-crafted feature representations [19], or access to symbolic representations [15]. It motivated the need for large-scale RPM datasets and the requirement for efficient reasoning models that utilized minimal prior knowledge. The first automatic RPM generation [28] work was based on using first-order logic, followed by two large-scale RPM datasets RAVEN [32] and Procedurally Generated Matrices (PGM) [2]. However, the RAVEN dataset contained a hidden shortcut solution where a model trained on the choice set only can achieve better performance than many state-of-the-art models. The reason behind this behavior is rooted in the creation of the choice set. Given the correct image, the distractor images were derived by randomly changing only one attribute. In response, two modified versions of the dataset, I-RAVEN by SRAN [11] and RAVEN-FAIR [3], were proposed to remove the shortcut solution and increase the difficulty levels of the distractors. Both the works, devised algorithm to generate a different set of distractors and provide evidence through experiments to claim the non-existence of any shortcut solution. The first significant advancement in RPM was by Wild Relational Network (WReN) [2], which applies the relation network of [24] multiple times to solve the abstract reasoning problem. LEN [34] learns to reason using a triplet of images in a row or column as input to a variant of the relation network. This work empirically supports improvements in performance using curriculum and reinforcement learning frameworks. CoPINet [33] suggests a contrastive learning algorithm to learn the underlying rules from given images. SRAN [11] designs a hierarchical rule-aware framework that learns rules through a series of steps of learning image representation, followed by row representation, and finally learning rules by pairing rows.

2.2 Transformer in Vision

Transformers [26] for machine translation have become widely adopted in numerous NLP tasks [7,22,14]. A transformer consists of self-attention layers added along with MLP layers. The self-attention mechanism plays a key role in drawing out the global dependencies between input and output. Grouped with the non-sequential processing of sentences, transformers demonstrate superiority in large-scale training scenarios compared to Recurrent Neural Networks. It avoids a drop in performance due to long-term dependencies. Recently, there has been

a steady influx of visual transformer models to various vision tasks: image classification [6], object detection [4], segmentation [29], image generation [21], video processing [31], VQA [13]. Among them, Vision Transformer (ViT) [8] designed for image classification has closed the gap with the performance provided by any state-of-the-art models (e.g., ImageNet, ResNet) based on convolution. Similar to the word sequence referred to as tokens required by transformers in NLP, ViT splits an image into patches and uses the linear embeddings of these patches as an input sequence. Our idea is closely related to learning intra-sequence relationships via self-attention.

3 Method

Before presenting our reasoning model, we provide a formal description of the RPM task in in Sec. 3.1 designed to measure abstract reasoning. The description articulates the condition required in a reasoning model to solve RPM questions successfully. In Sec. 3.2, we describe the three major components of our SAViR-T: the backbone network, the visual transformer, and the reasoning network. Our objective is based on using local feature maps as tokens for rule discovery among visual attributes along rows or columns to solve RPM questions.

3.1 Raven’s Progressive Matrices

Given a list of observed images in the form of Raven’s Progressive Matrix (\mathcal{M}) referred to as the context of size 3×3 with a missing final element at $\mathcal{M}_{3,3}$, where $\mathcal{M}_{i,j}$ denotes the j -th image at i -th row. In Figure 2, for ease of notation, we refer to the images in \mathcal{M} as index locations 1 through 16 as formatted in the dataset, where the first eight form the context matrix and the rest belong to the choice list. The task of a learner is to solve the context \mathcal{M} by finding the best-fit answer image from an unordered set of choices $\mathcal{A} = \{a_1, \dots, a_8\}$. The images in an RPM can be decomposed into attributes, objects, and the object count. Learning intra-relationship between these visual components will guide the model to form a stronger inter-relationship between images constrained by rules. The learner needs to locate objects in each image, extract their visual attributes such as color, size, and shape, followed by inferring the rules “ r ” such as “constant”, “progression”, “OR”, etc., that satisfies the attributes among a list of images. Usually, the rules “ r ” are applied row-wise or column-wise according to [5] on the decomposed visual elements. Since an RPM is based on a set of rules applied either row-wise or column-wise, the learner needs to pick the shared rules between the top two rows or columns. Among the choice list, the correct image from \mathcal{A} , when placed at the missing location in the last row or column, will satisfy these shared rules.

3.2 Our Approach: SAViR-T

Our method consists of three sub-modules: (i) a Backbone Network, (ii) a Visual Transformer (VT), and (iii) a Reasoning Network, trained end to end. Please

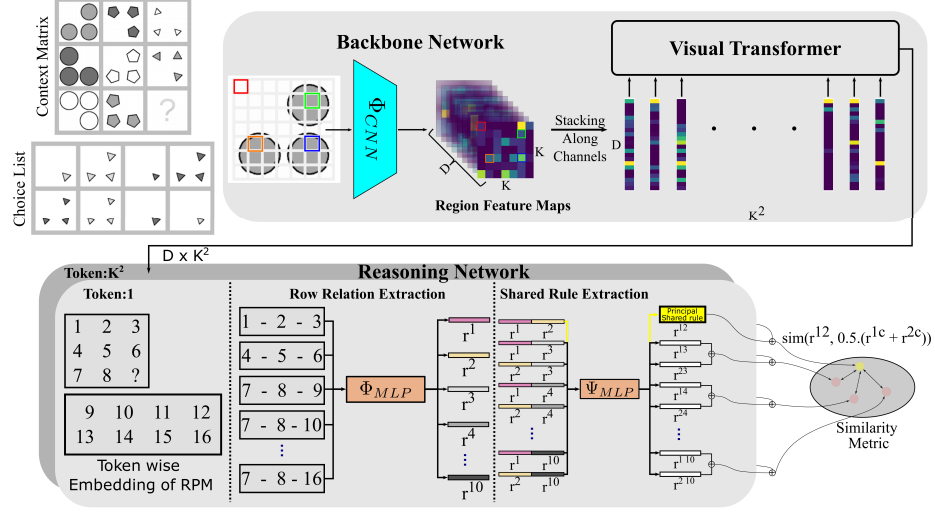


Fig. 2: SAViR-T consists of **Backbone Network**, **Visual Transformer**, **Reasoning Network**. Each image in \mathcal{M} given to the Backbone Network (Φ_{CNN}) to extract the “Feature Maps”, $f \in \mathbb{R}^{D \times K \times K}$. Visual Transformer attends on the features of local image patches and returns the attended vectors \hat{f} of each image. The Reasoning Network functions per patch (depicted as K parallel layers) for the entire context and choice attended vectors. Per patch, we start with group (row or column) rule extraction r^i via Φ_{MLP} , followed by shared Shared rule extraction r^{ij} via Ψ_{MLP} . The Principal Shared rule r^{12} is compared against extracted Shared rule for choice a , $\frac{1}{2}(r^{1c} + r^{2c})$ (“ c ” is the choice row index for choice image “ a ”). The choice image with max similarity score is predicted as the answer.

refer Figure 2 for an illustration of our model. First, we process the images in \mathcal{M} via several convolutional blocks referred to as the backbone network. The output feature map is given as to the visual transformer to extract the attended visual embedding. The attended embedding is given to the reasoning network to discover the embedded rule representation in RPM. A scoring function is used to rank the choice images by comparing its row or column representation (extracted by placing it in the missing location) with the rule representation and predicting the index with the highest similarity as the correct choice. We leverage the strength of convolutions, which learns location invariant low-level neighborhood structures and visual transformer to relate to the higher-order semantic concepts. We treat each local region as a token separately in reasoning and apply a fusion function to recover the hidden rules that point the model to the correct answer.

Backbone Network The backbone network receives as input an image from the context (\mathcal{M}) or choice list (\mathcal{A}) of size $\mathbb{R}^{C \times H \times W}$. The extracted feature map

(f_{ij}, f_a) is of dimension $\mathbb{R}^{D \times K \times K}$ where $K \times K$ is the number of image regions also referred to as tokens, and D is the dimension of the feature vector of each area. Accordingly, each feature vector corresponds to a $(\frac{W}{K} \times \frac{H}{K})$ pixel region retaining the spatial information of the raw image. We intend to summarize the high-level semantic information present in the image by learning from a set of low-level visual tokens. To this effect, we employ several convolutional blocks as our backbone network, denoted as Φ_{CNN} , to extract local features from images. We use ResNet [9] as our primary backbone network, although we show results with other popular backbones in our ablation study.

$$f_{ij} = \Phi_{\text{CNN}}(\mathcal{M}_{ij}), \quad i, j = 1, \dots, 3 \quad \text{and} \quad i, j \neq 3 \quad (1)$$

$$f_a = \Phi_{\text{CNN}}(a), \quad \forall a \in \mathcal{A} \quad (2)$$

Both the context and choice feature representation are generated using the same network. We flatten and concatenate the matrix format in context to prepare feature vector $F_{\mathcal{M}} = [f_{11}, \dots, f_{32}] \in \mathbb{R}^{8 \times K^2 \times D}$, where each feature map is reshaped into a K^2 -tall sequence of tokens, $\mathbb{R}^{K^2 \times D}$. Choices are processed in the same manner, $F_{\mathcal{A}} = [f_{a_1}, \dots, f_{a_8}] \in \mathbb{R}^{8 \times K^2 \times D}$. Finally, both the context and the choice representations are concatenated as $F = [F_{\mathcal{M}}, F_{\mathcal{A}}]$, with $[\cdot, \cdot]$ denoting the concatenation operator.

Visual Transformer To learn the concepts responsible for reasoning, we seek to model the interactions between local regions of an image as a bottom-up process, followed by top-down attention to encoder relations over the regions. We adopt Visual Transformer [8], which learns the attention weights between tokens to focus on relationally-relevant regions within images. The transformer is composed of "Multi-head Self-Attention" (MSA) mechanism followed by a multi-layer perceptron (MLP). Both are combined together in layers $l = 1, \dots, L$ to form the transformer encoder. A layer normalization layer (LN) and residual connection are added before and after every core component. The interactions between the tokens generate an attention map for each layer and head. Below we describe the steps involved in learning the attended vectors.

Reasoning Model Human representations of space are believed to be hierarchical [20], with objects parsed into parts and grouped into part constellations. To mimic this, SAViR-T generates weighted representations over local regions in the image described above. Our reasoning module combines the inductive bias in an RPM and per-patch representations to learn spatial relations between images. We start by translating these attended region vectors obtained from RPM above into within-group relational reasoning instructions, expressed in terms of the row representations r_k^i via Φ_{MLP} . These representations hold knowledge of the rules that bind the images in the i -th row. We realize ten row representations, including eight possible last rows, where each choice is replaced at the missing location (similarly for the columns). We define function Ψ_{MLP} which retrieves the common across-group rule representations r_k^{ij} , given the pair (r_k^i, r_k^j) . The

maximum similarity score between the last-row rules based on the Choice list, $\{r_k^{1c}, r_k^{2c}\}, \forall c = 3, \dots, 10$, and the extracted Principal Shared rule from the top two rows r_k^{12} indicates the correct answer.

Row Relation Extraction Module. Since our image encoding is prepared region-wise, we focus on these regions independently to detect patterns maintaining the order of RPM, i.e., row-wise (left to right) or column-wise (top to bottom). We restructure the resulting output of a RPM from the transformer \hat{F} as $\mathbb{R}^{K^2 \times 16 \times D}$, where the embeddings of each local region k for an image at index n in the RPM is denoted \hat{f}_k^n for $n = 1, 2, \dots, 16^1$ and $k = 1, \dots, K^2$. We collect these embeddings for every row and column as triplets $(\hat{f}_k^1, \hat{f}_k^2, \hat{f}_k^3), (\hat{f}_k^4, \hat{f}_k^5, \hat{f}_k^6), (\hat{f}_k^7, \hat{f}_k^8, \hat{f}_k^a), a = 9, \dots, 16$. Similarly, the triplets formed from columns are $(\hat{f}_k^1, \hat{f}_k^4, \hat{f}_k^7), (\hat{f}_k^2, \hat{f}_k^5, \hat{f}_k^8), (\hat{f}_k^3, \hat{f}_k^6, \hat{f}_k^a)$. Each row and column is concatenated along the feature dimension and processed through a relation extraction function Φ_{MLP} :

$$\begin{aligned} r_k^1 &= \Phi_{MLP}(\hat{f}_k^1, \hat{f}_k^2, \hat{f}_k^3), & c_k^1 &= \Phi_{MLP}(\hat{f}_k^1, \hat{f}_k^4, \hat{f}_k^7) \\ r_k^2 &= \Phi_{MLP}(\hat{f}_k^4, \hat{f}_k^5, \hat{f}_k^6), & c_k^2 &= \Phi_{MLP}(\hat{f}_k^2, \hat{f}_k^5, \hat{f}_k^8) \\ r_k^3 &= \Phi_{MLP}(\hat{f}_k^7, \hat{f}_k^8, \hat{f}_k^a), & c_k^3 &= \Phi_{MLP}(\hat{f}_k^3, \hat{f}_k^6, \hat{f}_k^a). \end{aligned} \quad (3)$$

The function Φ_{MLP} has two-fold aims: (a) it seeks to capture common properties of relational reasoning along a row/column for each local region; (b) it updates the attention weights to cast aside the irrelevant portions that do not contribute to rules.

Shared Rule Extraction Module. The common set of rules conditioned on visual attributes in the first two rows of a RPM is the Principal Shared rule set which the last row has to match to select the correct image in the missing location. The top two rows will contain these common sets of rules, possibly along with the rules unique to their specific rows. Given r_k^i, r_k^j , the goal of function Ψ_{MLP} is to extract these Shared rules between the pairs of rows/columns,

$$r_k^{ij} = \Psi_{MLP}(r_k^i, r_k^j), \quad c_k^{ij} = \Psi_{MLP}(c_k^i, c_k^j), \quad rc_k^{ij} = [r_k^{ij}, c_k^{ij}]. \quad (4)$$

Similar to Φ_{MLP} , function Ψ_{MLP} aims to elucidate the Shared relationships between any pair i, j of rows/columns. Those relationships should hold across image patches, thus we fuse rc_k^{ij} obtained for $k = 1, \dots, K^2$ regions via averaging, leading to the Shared rule embedding $rc^{ij} = \frac{1}{K^2} \sum_k rc_k^{ij}$ for the entire image.

3.3 Training and Inference

Given the principal Shared rule embedding rc^{12} from the top two row pairs, a similarity metric is a function of closeness between $\text{sim}(rc^{12}, rc^a)$, where $rc^a = \frac{1}{2}(rc^{1c} + rc^{2c}), \forall c = 3, \dots, 10$ is the average of the Shared rule embeddings among the choice a and the top two rows, and $\text{sim}(\cdot, \cdot)$ is the inner product between

¹ $16 = 8 + 8$ for eight Context and eight Choice images of an RPM.

rc^{12} and the average Shared rule embedding. The similarity score will be higher for the correct image is placed at the last row compared to the wrong choice,

$$a^* = \arg \max_a \text{sim}(rc^{12}, rc^a). \quad (5)$$

We use cross-entropy as our loss function to train **SAViR-T** end-to-end. To bolster generalization property of our model, two types of augmentation were adapted from [33]: (i) shuffle the order of the top two rows or columns, as the resulting change will not affect the final solution since the rules remain unaffected; and (ii) shuffling the index location of the correct image in the unordered set of choice list. After training our model, we can use **SAViR-T** to solve new RPM problems (i.e., during testing) by applying (5).

4 Experiments

We study the effectiveness of our proposed **SAViR-T** for solving the challenging RPM questions, specifically focusing on PGM [2], RAVEN [32], I-RAVEN [11], RAVEN-FAIR [3], and V-PROM [25]. Details about the datasets can be found in the Supplementary. Next, we describe the experimental details of our simulations, followed by the results of these experiments and the performance analysis of obtained results, including an ablation study of our **SAViR-T**.

4.1 Experimental Settings

We trained **SAViR-T** for 100 epochs on all three RAVEN datasets, 50 epochs for PGM and 100 epochs for V-PROM, where each RPM is scaled to $16 \times 224 \times 224$. For V-PROM, similar to [25], we use the features extracted from the pre-trained ResNet-101 before the last pooling layer; i.e., dimension of $2048 \times 7 \times 7$. To further reduce the complexity in case of V-PROM, we use an MLP layer to derive $512 \times 7 \times 7$ feature vectors; we must mention that this MLP becomes part of **SAViR-T**'s training parameters. We use the validation set to track model performance during the training process and use the best validation checkpoint to report the accuracy on the test set. For **SAViR-T**, we adopt ResNet-18 as our backbone for results in Table 1 and Table 2. Our transformer depth is set to one for all datasets and the counts of heads set to 3 for RAVEN datasets, to 6 for PGM and to 5 for V-PROM. Finally, in our reasoning module, we use a two-layer MLP, Φ_{MLP} , and a four-layer MLP, Ψ_{MLP} with a dropout of 0.5 applied to the last layer. As the RAVEN dataset is created by applying rules row-wise, we set the column vector c_k^{ij} in (4) as zero-vector while training our model. No changes are made while training for PGM, as the tuple (rule, object, attribute) can be applied either along the rows or columns.

4.2 Performance Analysis

Table 1 summarizes the performance of our model and other baselines on the test set of RAVEN and I-RAVEN datasets. We report the scores from [35] for

Table 1: Model performance (%) on RAVEN / I-RAVEN.

Method	Acc	Center	2×2Grid	3×3Grid	L-R	U-D	O-IC	O-IG
LSTM [10]	13.1/18.9	13.2/26.2	14.1/16.7	13.7/15.1	12.8/14.6	12.4/16.5	12.2/21.9	13/21.1
WReN [2]	34.0/23.8	58.4/29.4	38.9/26.8	37.7/23.5	21.6/21.9	19.7/21.4	38.8/22.5	22.6/21.5
ResNet+DRT [32]	59.6/40.4	58.1/46.5	46.5/28.8	50.4/27.3	65.8/50.1	67.1/49.8	69.1/46.0	60.1/34.2
LEN [34]	72.9/39.0	80.2/45.5	57.5/27.9	62.1/26.6	73.5/44.2	81.2/43.6	84.4/50.5	71.5/34.9
CoPINet [33]	91.4/46.1	95.1/54.4	77.5/36.8	78.9/31.9	99.1/51.9	99.7/52.5	98.5/52.2	91.4/42.8
SRAN [11]	56.1 [†] /60.8	78.2 [†] /78.2	44.0 [†] /50.1	44.1 [†] /42.4	65.0 [†] /70.1	61.0 [†] /70.3	60.2 [†] /68.2	40.1 [†] /46.3
DCNet [36]	93.6/49.4	97.8/57.8	81.7/34.1	86.7 /35.5	99.8 /58.5	99.8 /60	99.0 /57.0	91.5 /42.9
SCL [30]	91.6/95.0	98.1 /99.0	91.0/96.2	82.5/89.5	96.8/97.9	96.5/97.1	96.0/97.6	80.1/87.7
SAViR-T (Ours)	94.0 / 98.1	97.8/ 99.5	94.7 / 98.1	83.8/ 93.8	97.8/ 99.6	98.2/ 99.1	97.6/ 99.5	88.0/ 97.2
Human	84.41/-	95.45/-	81.82/-	79.55/-	86.36/-	81.81/-	86.36/-	81.81/-

[†] indicates our evaluation of the baseline in the absence of published results.

Table 2: Test accuracy of different models on PGM.

	LSTM [10]	ResNet	CoPINet [33]	WReN [2]	MXGNet [27]	LEN [34]	SRAN [11]	DCNet [36]	SCL [30]	SAViR-T (Ours)
Acc	35.8	42.0	56.4	62.6	66.7	68.1	71.3	68.6	88.9	91.2

Table 3: Test accuracy of different models on V-PROM.

Method	RN [25]	DCNet [36]	SRAN [11]	SAViR-T (Ours)
Acc	52.8 ^{††}	30.4 [†]	40.8 [†]	62.6

[†] indicates our evaluation of the baseline in the absence of published results.

^{††} indicates our evaluation of the baseline (reported result in [25] was 51.2).

I-RAVEN on LEN, COPINet, and DCNet. We also report the performance of humans on the RAVEN dataset [32]; there is no reported human performance on I-RAVEN. Overall, our SAViR-T achieves superior performance among all baselines for I-RAVEN and a strong performance, on average, on RAVEN. Our method performs similar to DCNet for RAVEN with a slight improvement of 0.4%. We notice DCNet has better performance over ours by a margin of 1.4% – 3.5% over “3 × 3 Grid”, “L-R”, “U-D”, “O-IC” and “O-IG”, while we show 13% improvement for “2 × 2 Grid”. For I-RAVEN, the average test accuracy of our model improves from 95% (SCL) to 98.13% and shows consistent improvement over all configurations across all models. The most significant gain, spotted in “3 × 3 Grid” and

Table 4: Test accuracy of different models on RAVEN-FAIR.

Method	ResNet [9]	LEN [34]	COPINet [33]	DCNet [36]	MRNet [3]	SAViR-T (Ours)
Acc	72.5	78.3	91.4	54.5 [†]	96.6	97.4

[†] indicates our evaluation of the baseline in the absence of published results.

“O-IG” is expected since our method learns to attend to semantic spatio-visual tokens. As a result, we can focus on the smaller scale objects present in these configurations, which is essential since the attributes of these objects define the rules of the RPM problem. In Table 1, DCNet achieves better accuracy for “3 × 3”, “L-R”, “U-D”, “O-IC”, and “O-IG” for RAVEN but significantly lower accuracy for I-RAVEN, **suggesting DCNet exploits the shortcut in RAVEN**. See our analysis in Table 5 that supports this observation.

We also observe that our reasoning model shows more significant improvements on I-RAVEN than RAVEN. This is because the two datasets differ in the selection process of the negative choice set. The wrong images differ in only a single attribute from the right panel for RAVEN, while in I-RAVEN, they differ in at least two characteristics. The latter choice set prevents models from deriving the puzzle solution by only considering the available choices. At the same time, this strategy also helps the classification problem (better I-RAVEN scores) since now the choice set images are more distinct. In Table 4, we also report the test scores on the RAVEN-FAIR dataset against several baselines. Similar to the above, our model achieves the best performance.

Table 2 reports performance of SAViR-T and other models trained on the neutral configuration in the PGM dataset. Our model improves by 2.3% over the best baseline model (SCL). PGM dataset is 20 times larger than RAVEN, and the applied rule can be present either row-wise or column-wise. Additionally, PGM contains “line” as an object type, increasing the complexity compared to RAVEN datasets. Even under these additional constraints, SAViR-T is able to improve RPM solving by mimicking the reasoning process.

In Table 3 we report the performance on the V-PROM dataset, made up of natural images. The background signal for every image in the dataset can be considered as noise or a distractor. In this highly challenging benchmark, our SAViR-T shows a major improvement of over 8% over the Relation Network (RN) reported in [25] (51.2 reported in [25] and 52.83 for our evaluation of the RN model –since the code is not available–). Since the V-PROM dataset is the most challenging one, we define the margin Δ for each testing sample to better understand the performance of SAViR-T:

$$\Delta = \text{sim}(rc^{12}, rc^{a^*}) - \max_{a \neq a^*} \text{sim}(rc^{12}, rc^a),$$

where a^* indicates the correct answer among eight choices. The model is confident and correctly answers the RPM question for $\Delta \gg 0$. When $\Delta \approx 0$, the

model is uncertain about its predictions, $\Delta < 0$ indicating incorrect and $\Delta > 0$ correct uncertain predictions. Finally, the model is confident but incorrectly answers for $\Delta \ll 0$.

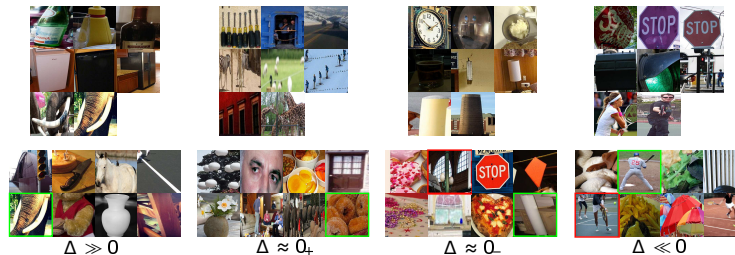


Fig. 3: V-PROM RPM examples (trained on SAViR-T) for 4-cases of Δ in order from correct prediction with strong certainty to incorrect prediction with strong certainty.

In Figure 3 we present examples from the V-PROM testing set. In the Supplementary we present additional detailed analysis of these results. The first example has a $\Delta > 100$, which means that the trained SAViR-T is very confident about its prediction. Next, we move to examples with Δ very close to zero, either negative or positive. We visualize two such examples in Figure 3, in the second (correctly classified RPM) and the third (misclassified) puzzles. The second is a counting problem where the first image in a row has x objects and the next two images in the same row y (i.e., first row $x = 7, y = 2$). Some images are distorted and/or blurred after the pre-processing required to use the pre-trained ResNet-101, which makes the recognition and “counting” of objects difficult. The second example is an “And” rule on object attributes. The first row contains circle objects and the second and third cylinders. The wrongly selected image (depicted in a red bounding box) contains as well cylinder objects. In these two cases, the model is uncertain about which image is solving the RPM.

Lastly, we study examples which SAViR-T very confidently misclassifies. Specifically, we picked the worst six instances of the testing set. One of these examples is depicted in the last puzzle of Figure 3. All examples belong to the “And rule” for object attributes like the case $\Delta \gg 0$. In the depicted example, “And” rule of the last row refers to “Players”; where in both cases, they are playing “Tennis”. The problem is that images five and eight depict “Players” playing “Tennis”, resulting in a controversial situation. Although the model misclassifies the “Player” in the second choice, it is reasonable to choose either the second, fifth, or eighth images as the correct image in the puzzle.

Table 5 presents our cross-dataset results between models trained and tested on different RAVEN-based datasets. Since all three datasets only differ in the manner their distractors in the Choice set were created but are identical in the Context, a model close to the generative process should be able to pick the correct image irrespective of how difficult the distractors are. In the first two

Table 5: Results of cross-dataset evaluation. Top row indicates the training set, next the test set.

Model Training Set	RAVEN		I-RAVEN		RAVEN-F	
Evaluation Dataset	I-RAVEN	RAVEN-F	RAVEN	RAVEN-F	RAVEN	I-RAVEN
SRAN	72.8	78.5	57.1	71.9	54.7	60.5
DCNet	14.7	27.4	37.9	51.7	57.8	46.5
SAViR-T (Ours)	97	97.5	95.1	97.7	94.7	88.3

columns, we train SAViR-T with the RAVEN dataset and measure the trained model performance on I-RAVEN, RAVEN-FAIR (RAVEN-F) testing sets; other combinations in the succeeding columns follow the respective train-test patterns. As was expected, when training on RAVEN (94%), we see an improvement on both I-RAVEN (97%) and RAVEN-FAIR (97.5%) test since the latter datasets have more dissimilar choice images, helping the reasoning problem. This increase in performance can be seen for SRAN from 56.1% to 72.8% and 78.5% respectively. Since DCNet (93.6%) utilizes the short solution, the accuracy drops to 14.7% and 27.4% respectively. For the same reasons, when training on I-RAVEN (98.8%), our model shows a drop in RAVEN (95.1%) performance; for RAVEN-FAIR (97.7%), the performance remains close to the one on the training dataset.

4.3 Ablation Study

Exploiting shortcut solutions As shown in SRAN [11], any powerful model that learns by combining the extracted features from the choices is capable of exploiting the shortcut solution present in the original RAVEN. In a context-blind setting, a model trained only on images in the RAVEN choice list should predict randomly. However, context-blind {ResNet, CoPINet, DCNet} models attain 71.9%, 94.2% and 94.1% test accuracy respectively. We train and report the accuracy for context-blind DCNet and reported the scores in [11] for ResNet, CoPINet. Similarly, we investigate our SAViR-T in a context-blind setting. We remove the reasoning module and use the extracted attended choice vectors from the visual transformer, passed through an MLP, to output an eight-dimensional logit vector. After training for 100 epochs, our model performance remained at 12.2%, similar to the random guess of $1/8 = 12.5\%$, suggesting that our Backbone with the Visual Transformer does not contribute towards finding a shortcut. Thus, our semantic tokenized representation coiled with the reasoning module learns rules from the context to solve the RPM questions.

Does SAViR-T learn rules? The rules in RPMs for the I-RAVEN dataset are applied row-wise. However, these rules can exist in either rows or columns. We evaluate performance on two different setups to determine if our model can discover the rule embeddings with no prior knowledge of whether the rules were applied row-wise or column-wise. In our first setup, we train SAViR-T with the

prior knowledge of row-wise rules in RPMs. We train our second model by preparing rule embeddings on both row and column and finally concatenating them to predict the correct answer. Our model performance drop was only 3.8% from 98.1% to 94.3%. In the case of SRAN [11], the reported drop in performance was 1.2% for the above setting. Overall, this indicates that our model is capable of ignoring the distraction from the column-wise rule application.

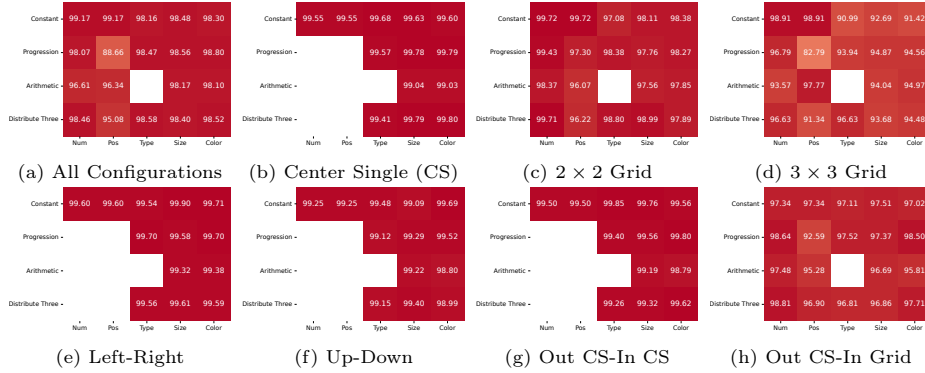


Fig. 4: Testing set classification accuracy for the SAViR-T trained on all configurations based on each I-RAVEN rule (constant, progression, arithmetic, and distribute three) and used attribute (number, position, type, size, color of the objects). From left to right, Figure 4a presents the classification accuracy for all configurations, Figure 4b for “Center Single”, Figure 4c for 2×2 Grid, and Figure 4d for 3×3 Grid in the top row. Similarly in the bottom row, Figure 4e for left right, Figure 4f for the up down, 4g for out single, in center single and finally Figure 4h for out single, in 2×2 Grid.

Figure 4 presents the I-RAVEN performance on the test set for SAViR-T when trained on the I-RAVEN dataset. The eight different heatmap images correspond to the setting with all configurations (Figure 4a) and individual configurations from “Center Single”(Figure 4b) to “Out-In-Grid” (Figure 4h). The row dimension in each heatmap corresponds to the RPM rules used in the puzzles (Constant, Progression, Arithmetic, and Distribute Three). In I-RAVEN, each rule is associated with an attribute. Therefore, in the columns, we identify the characteristics of different objects, such as their “Number,” “Position,” “Type,” “Size,” and “Color”. The Blank cells in the heatmap e.g., “(Arithmetic, Type)”, indicate non-existence for that (rule, attribute) pair combination in the dataset.

The most challenging combination of (rule, attribute) is a progression with the position. In this setup, the different objects progressively change position on the 2×2 and 3×3 grids. The models fail to track this change well. To further understand this drop in performance, we performed extended experiments, investigating the difference in attributes between the correct image and the predicted (misclassified) image (more details in the Appendix). We notice that for the 2×2 grid configuration, the predicted differences are only in one

attribute, primarily the “Position”. This means the distractor objects are of the same color, size, type, and number as the correct one but have different positions inside the grid. The second group of misclassified examples differs in the “Type” of the present objects. Therefore, we can conclude the model finds it challenging to track the proper position of the entities and their type for some examples. Again, in the 3×3 grid, “Position” is the main differentiating attribute, but the “Size” attribute follows it; this makes sense since, in the 3×3 grid, the objects have a petite size resulting in greater sensitivity to distinguish the different scales. Similar behavior is observed in the O-IG configuration.

5 Conclusions

In this paper, we introduced SAViR-T, a model that takes into account the visually-critical spatial context present in image-based RPM. By partitioning an image into patches and learning relational reasoning over these local windows, our SAViR-T fosters the learning of Principal rule and attribute representations in RPM. The model recognizes the Principal Shared rule, comparing it to choices via a simple similarity metric, thus avoiding the possibility of finding a shortcut solution. SAViR-T shows robustness to injection of triplets that disobey the RPM formation patterns, e.g., when trained with both choices of row- and column-wise triplets on the RPM with uniquely, but unknown, row-based rules. We are the first to provide extensive experiments results on all three RAVEN-based datasets, PGM, and the challenging natural image-based V-PROM, which suggests that SAViR-T outperforms all baselines by a significant margin except for RAVEN where we match their accuracy.

References

1. Apps, J.N.: Abstract thinking. Encyclopedia of Aging and Public Health. Springer Link (2008)
2. Barrett, D., Hill, F., Santoro, A., Morcos, A., Lillicrap, T.: Measuring abstract reasoning in neural networks. In: International conference on machine learning. pp. 511–520. PMLR (2018)
3. Benny, Y., Pekar, N., Wolf, L.: Scale-localized abstract reasoning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12557–12565 (2021)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Carpenter, P.A., Just, M.A., Shell, P.: What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. Psychological review **97**(3), 404 (1990)
6. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International Conference on Machine Learning. pp. 1691–1703. PMLR (2020)

7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020)
9. 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)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural computation* **9**(8), 1735–1780 (1997)
11. Hu, S., Ma, Y., Liu, X., Wei, Y., Bai, S.: Stratified rule-aware network for abstract visual reasoning. *arXiv preprint arXiv:2002.06838* (2020)
12. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. *Behavioral and brain sciences* **40** (2017)
13. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019)
14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019)
15. Lovett, A., Forbus, K., Usher, J.: Analogy with qualitative spatial representations can simulate solving raven’s progressive matrices. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. vol. 29 (2007)
16. Lovett, A., Forbus, K., Usher, J.: A structure-mapping model of raven’s progressive matrices. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. vol. 32 (2010)
17. Lovett, A., Tomai, E., Forbus, K., Usher, J.: Solving geometric analogy problems through two-stage analogical mapping. *Cognitive science* **33**(7), 1192–1231 (2009)
18. Małkiński, M., Mańdziuk, J.: A review of emerging research directions in abstract visual reasoning. *arXiv preprint arXiv:2202.10284* (2022)
19. McGregor, K., Goel, A.: Confident reasoning on raven’s progressive matrices tests. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 28 (2014)
20. Palmer, S.E.: Hierarchical structure in perceptual representation. *Cognitive psychology* **9**(4), 441–474 (1977)
21. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: *International Conference on Machine Learning*. pp. 4055–4064. PMLR (2018)
22. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding with unsupervised learning (2018)
23. Raven, J.C., Court, J.H.: Raven’s progressive matrices and vocabulary scales, vol. 759. Oxford psychologists Press Oxford (1998)
24. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. *Advances in neural information processing systems* **30** (2017)
25. Teney, D., Wang, P., Cao, J., Liu, L., Shen, C., van den Hengel, A.: V-prom: A benchmark for visual reasoning using visual progressive matrices. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 34, pp. 12071–12078 (2020)

26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
27. Wang, D., Jamnik, M., Lio, P.: Abstract diagrammatic reasoning with multiplex graph networks. *arXiv preprint arXiv:2006.11197* (2020)
28. Wang, K., Su, Z.: Automatic generation of raven’s progressive matrices. In: *Twenty-fourth international joint conference on artificial intelligence* (2015)
29. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 8741–8750 (2021)
30. Wu, Y., Dong, H., Grosse, R., Ba, J.: The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *arXiv preprint arXiv:2007.04212* (2020)
31. Zeng, Y., Fu, J., Chao, H.: Learning joint spatial-temporal transformations for video inpainting. In: *European Conference on Computer Vision*. pp. 528–543. Springer (2020)
32. 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/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5317–5327 (2019)
33. Zhang, C., Jia, B., Gao, F., Zhu, Y., Lu, H., Zhu, S.C.: Learning perceptual inference by contrasting. *Advances in Neural Information Processing Systems* **32** (2019)
34. Zheng, K., Zha, Z.J., Wei, W.: Abstract reasoning with distracting features. *Advances in Neural Information Processing Systems* **32** (2019)
35. Zhuo, T., Huang, Q., Kankanhalli, M.: Unsupervised abstract reasoning for raven’s problem matrices. *IEEE Transactions on Image Processing* **30**, 8332–8341 (2021)
36. Zhuo, T., Kankanhalli, M.: Effective abstract reasoning with dual-contrast network. In: *International Conference on Learning Representations* (2020)