

RAVEN: A Dataset for Relational and Analogical Visual Reasoning

Chi Zhang^{1,2} Feng Gao^{1,2} Baoxiong Jia¹ Yixin Zhu^{1,2} Song-Chun Zhu^{1,2}
¹ UCLA Center for Vision, Cognition, Learning and Autonomy
² International Center for AI and Robot Autonomy (CARA)
 {chi.zhang, f.gao, baoxiongjia, yixin.zhu}@ucla.edu, sczhu@stat.ucla.edu

Abstract

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

1. Introduction

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

— David Marr, 1982 [35]

Computer vision has a wide spectrum of tasks. Some computer vision problems are clearly purely visual, “cap-

indicates equal contribution.

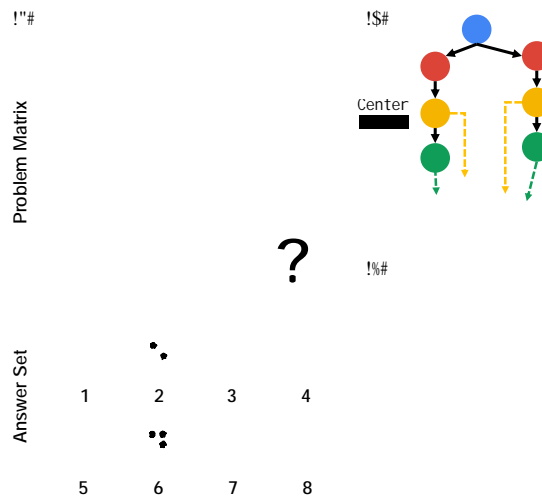


Figure 1. (a) An example RPM. One is asked to select an image that best completes the problem matrix, following the structural and analogical relations. Each image has an underlying structure. (b) Specifically in this problem, it is an inside-outside **structure** in which the outside **component** is a **layout** with a single centered object and the inside **component** is a 2×2 grid **layout**. Details in Figure 2. (c) lists the rules for (a). The compositional nature of the rules makes this problem a difficult one. The correct answer is 7.

turing” the visual information process; for instance, filters in early vision [5], primal sketch [13] as the intermediate representation, and Gestalt laws [24] as the perceptual organization. In contrast, some other vision problems have trivialized requirements for perceiving the image, but engage more generalized problem-solving in terms of relational and/or analogical visual reasoning [16]. In such cases, the vision component becomes the “basis for decisions about our thoughts and actions”.

Currently, the majority of the computer vision tasks focus on “capturing” the visual information process; few lines of work focus on the later part—the relational and/or analogical visual reasoning. One existing line of work in equipping artificial systems with reasoning ability hovers around Visual Question Answering (VQA) [2, 22, 48, 58, 62]. However, the reasoning skills required in VQA lie only at the periphery of the cognitive ability test circle [7]. To

push the limit of computer vision or more broadly speaking, Artificial Intelligence (AI), towards the center of cognitive ability test circle, we need a test originally designed for measuring human’s intelligence to challenge, debug, and improve the current artificial systems.

A surprisingly effective ability test of human visual reasoning has been developed and identified as the Raven’s Progressive Matrices (RPM) [28, 47, 52], which is widely accepted and believed to be highly correlated with real intelligence [7]. Unlike VQA, RPM lies directly at the center of human intelligence [7], is diagnostic of abstract and structural reasoning ability [9], and characterizes the defining feature of high-level intelligence, *i.e.*, *fluid intelligence* [21].

Figure 1 shows an example of RPM problem together with its structure representation. Provided two rows of figures consisting of visually simple elements, one must efficiently derive the correct image structure (Figure 1(b)) and the underlying rules (Figure 1(c)) to jointly reason about a candidate image that best completes the problem matrix. In terms of levels of reasoning required, RPM is arguably harder compared to VQA:

- Unlike VQA where natural language questions usually imply what to pay attention to in the image, RPM relies merely on visual clues provided in the matrix and the *correspondence problem* itself, *i.e.*, finding the correct level of attributes to encode, is already a major factor distinguishing populations of different intelligence [7].
- While VQA only requires spatial and semantic understanding, RPM needs joint spatial-temporal reasoning in the problem matrix and the answer set. The limit of *short-term memory*, the ability of *analogy*, and the discovery of the *structure* have to be taken into consideration.
- Structures in RPM make the compositions of rules much more complicated. Unlike VQA whose questions only encode relatively simple first-order reasoning, RPM usually includes more sophisticated logic, even with recursions. By composing different rules at various levels, the reasoning progress can be extremely difficult.

To push the limit of current vision systems’ reasoning ability, we generate a new dataset to promote further research in this area. We refer to this dataset as the Relational and Analogical Visual Reasoning dataset (RAVEN) in homage to John Raven for the pioneering work in the creation of the original RPM [47]. In summary:

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

The RAVEN dataset is designed inherently to be light

in visual recognition and heavy in reasoning. Each image only contains a limited set of simple gray-scale objects with clear-cut boundaries and no occlusion. In the meantime, rules are applied row-wise, and there could be one rule for each attribute, attacking visual systems’ major weaknesses in *short-term memory* and *compositional reasoning* [22].

An obvious paradox is: in this innately compositional and structured RPM problem, no annotations of structures are available in previous works (*e.g.*, [3, 55]). Hence, we set out to establish a semantic link between visual reasoning and structure reasoning in RPM. We ground each problem instance to a sentence derived from an Attributed Stochastic Image Grammar (A-SIG) [12, 30, 43, 56, 60, 61] and decompose the data generation process into two stages: the first stage samples a sentence from a pre-defined A-SIG and the second stage renders an image based on the sentence. This structured design makes the dataset very diverse and easily extendable, enabling generalization tests in different figure configurations. More importantly, the data generation pipeline naturally provides us with abundant dense annotations, especially the structure in the image space. This semantic link between vision and structure representation opens new possibilities by breaking down the problem into image understanding and tree- or graph-level reasoning [26, 53]. As shown in Section 6, we empirically demonstrate that models with a simple structure reasoning module to incorporate both vision-level understanding and structure-level reasoning would notably improve their performance in RPM.

The organization of the paper is as follows. In Section 2, we discuss related work in visual reasoning and computational efforts in RPM. Section 3 is devoted to a detailed description of the RAVEN dataset generation process, with Section 4 benchmarking human performance and comparing RAVEN with a previous RPM dataset. In Section 5, we propose a simple extension to existing models that incorporates vision understanding and structure reasoning. All baseline models and the proposed extensions are evaluated in Section 6. The notable gap between human subjects (84%) and vision systems (59%) calls for further research into this problem. We hope RAVEN could contribute to the long-standing effort in human-level reasoning AI.

2. Related Work

Visual Reasoning Early attempts were made in 1940s-1970s in the field of logic-based AI. Newell argued that one of the potential solutions to AI was “to construct a single program that would take a standard intelligence test” [42]. There are two important trials: (i) Evans presented an AI algorithm that solved a type of geometric analogy tasks in the Wechsler Adult Intelligence Scale (WAIS) test [10, 11], and (ii) Simon and Kotovsky devised a program that solved Thurstone letter series completion problems [54]. However, these early attempts were heuristic-based with hand-crafted rules, making it difficult to apply to other problems.

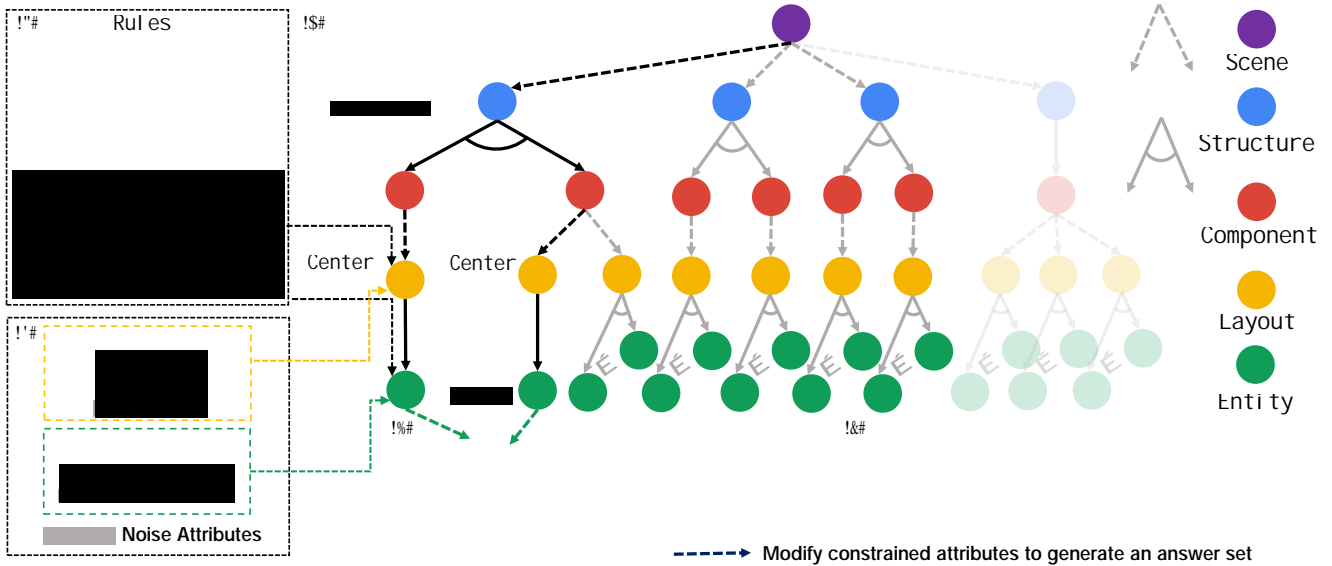


Figure 2. RAVEN creation process. A graphical illustration of the grammar production rules used in A-SIG is shown in (b). Note that **Layout** and **Entity** have associated attributes (c). Given a randomly sampled rule combination (a), we first prune the grammar tree (the transparent branch is pruned). We then sample an image structure together with the values of the attributes from (b), denoted by black, and apply the rule set (a) to generate a single row. Repeating the process three times yields the entire problem matrix in (d). (e) Finally, we sample constrained attributes and vary them in the correct answer to break the rules and obtain the candidate answer set.

The reasoning ability of modern vision systems was first systematically analyzed in the CLEVR dataset [22]. By carefully controlling inductive bias and slicing the vision systems’ reasoning ability into several axes, Johnson *et al.* successfully identified major drawbacks of existing models. A subsequent work [23] on this dataset achieved good performance by introducing a program generator in a structured space and combining it with a program execution engine. A similar work that also leveraged language-guided structured reasoning was proposed in [18]. Modules with special attention mechanism were latter proposed in an end-to-end manner to solve this visual reasoning task [19, 49, 59]. However, superior performance gain was observed in very recent works [6, 36, 58] that fell back to structured representations by using primitives, dependency trees, or logic. These works also inspire us to incorporate structure information into solving the RPM problem.

More generally, Bisk *et al.* [4] studied visual reasoning in a 3D block world. Perez *et al.* [46] introduced a conditional layer for visual reasoning. Aditya *et al.* [1] proposed a probabilistic soft logic in an attention module to increase model interpretability. And Barrett *et al.* [3] measured abstract reasoning in neural networks.

Computational Efforts in RPM The research community of cognitive science has tried to attack the problem of RPM with computational models earlier than the computer science community. However, an oversimplified assumption was usually made in the experiments that the computer programs had access to a symbolic representation of the image and the operations of rules [7, 32, 33, 34]. As reported in Section 4.4, we show that giving this critical information essentially turns it into a searching prob-

lem. Combining it with a simple heuristics provides us an optimal solver, easily surpassing human performance. Another stream of AI research [31, 37, 38, 39, 50] tries to solve RPM by various measurements of image similarity. To promote fair comparison between computer programs and human subjects in a data-driven manner, Wang and Su [55] first proposed a systematic way of automatically generating RPM using first-order logic. Barrett *et al.* [3] extended their work and introduced the Procedurally Generating Matrices (PGM) dataset by instantiating each rule with a relation-object-attribute tuple. Hoshen and Werman [17] first trained a CNN to complete the rows in a simplistic evaluation environment, while Barrett *et al.* [3] used an advanced Wild Relational Network (WReN) and studied its generalization.

3. Creating RAVEN

Our work is built on prior work aforementioned. We implement all relations in Advanced Raven’s Progressive Matrices identified by Carpenter *et al.* [7] and generate the answer set following *the monotonicity of RPM’s constraints* proposed by Wang and Su [55].

Figure 2 shows the major components of the generation process. Specifically, we use the A-SIG as the representation of RPM; each RPM is a parse tree that instantiates from the A-SIG. After rules are sampled, we prune the grammar to make sure the relations could be applied on any sentence sampled from it. We then sample a sentence from the pruned grammar, where rules are applied to produce a valid row. Repeating such a process three times yields a problem matrix. To generate the answer set, we modify attributes on the correct answer such that the relationships are broken.

Figure 3. Examples of RPM that show the effects of adding *noise* attributes. (Left) **Position**, **Type**, **Size**, and **Color** could vary freely as long as **Number** follows the rule. (Right) **Position** and **Type** in the inside group could vary freely.

Finally, the structured presentation is fed into a rendering engine to generate images. We elaborate the details below¹.

3.1. Defining the Attributed Grammar

We adopt an A-SIG as the hierarchical and structured image grammar to represent the RPM problem. Such representation is advanced compared with prior work (e.g., [3, 55]) which, at best, only maintains a flat representation of rules.

See Figure 2 for a graphical illustration of the grammar production rules. Specifically, the A-SIG for RPM has 5 levels—**Scene**, **Structure**, **Component**, **Layout**, and **Entity**. Note that each grammar level could have multiple instantiations, i.e., different categories or types. The **Scene** level could choose any available **Structure**, which consists of possibly multiple **Components**. Each **Component** branches into **Layouts** that links **Entities**. Attributes are appended to certain levels; for instance, (i) **Number** and **Position** are associated with **Layout**, and (ii) **Type**, **Size**, and **Color** are associated with **Entity**. Each attribute could take a value from a finite set. During sampling, both image structure and attribute values are sampled.

To increase the challenges and difficulties in the RAVEN dataset, we further append 2 types of *noise* attributes—**Uniformity** and **Orientation**—to **Layout** and **Entity**, respectively. **Uniformity**, set false, will not constrain **Entities** in a **Layout** to look the same, while **Orientation** allows an **Entity** to self-rotate. See Figure 3 for the effects of the noise attributes.

This grammatical design of the image space allows the dataset to be very diverse and easily extendable. In this dataset, we manage to derive 7 configurations by combining different **Structures**, **Components**, and **Layouts**. Figure 4 shows examples in each figure configuration.

¹See the supplementary material for production rules, semantic meanings of rules and nodes, and more examples.

3.2. Applying Rules

Carpenter *et al.* [7] summarized that in the advanced RPM, rules were applied row-wise and could be grouped into 5 types. Unlike Berrett *et al.* [3], we strictly follow Carpenter *et al.*’s description of RPM and implement all the rules, except that we merge **Distribute Two** into **Distribute Three**, as the former is essentially the latter with a null value in one of the attributes.

Specifically, we implement 4 types of rules in RAVEN: **Constant**, **Progression**, **Arithmetic**, and **Distribute Three**. Different from [3], we add internal parameters to certain rules (e.g., **Progression** could have increments or decrements of 1 or 2), resulting in a total of 8 distinct rule instantiations. Rules do not operate on the 2 noise attributes. As shown in Figure 1 and 2, they are denoted as [attribute: rule] pairs.

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

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

3.3. Generating the Answer Set

To generate the answer set, we first derive the correct representation of the solution and then leverage the monotonicity of RPM constraints proposed by Wang and Su [55]. To break the correct relationships, we find an attribute that is constrained by a rule as described in Section 3.2 and vary it. By modifying only one attribute, we could greatly reduce the computation. Such modification also increases the difficulty of the problem, as it requires attention to subtle difference to tell an incorrect candidate from the correct one.

4. Comparison and Analysis

In this section, we compare RAVEN with the existing PGM, presenting its key features and some statistics in Section 4.1. In addition, we fill in two missing pieces in a desirable RPM dataset, i.e., structure and hierarchy (Section 4.2), as well as the human performance (Section 4.3). We also show that RPM becomes trivial and could be solved instantly using a heuristics-based searching method (Section 4.4), given a symbolic representation of images and operations of rules.

4.1. Comparison with PGM

Table 1 summarizes several essential metrics of RAVEN and PGM. Although PGM is larger than RAVEN in terms of size, it is very limited in the average number of rules (**AvgRule**), rule instantiations (**RuleIns**), number of struc-

Figure 4. Examples of 7 different figure configurations in the proposed RAVEN dataset.

tures (**Struct**), and figure configurations (**FigConfig**). This contrast in PGM’s gigantic size and limited diversity might disguise model fitting as a misleading reasoning ability, which is unlikely to generalize to other scenarios.

Table 1. Comparison with the PGM dataset.

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

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

4.2. Introduction of Structure

A distinctive feature of RAVEN is the introduction of the structural representation of the image space. Wang and Su [55] and Barrett *et al.* [3] used plain logic and flat rule representations, respectively, resulting in no base of the structure to perform reasoning on. In contrast, we have in total 1,120,000 structure annotations (**StructAnno**) in the form of parsed sentences in the dataset, pairing each problem instance with 16 sentences for both the matrix and the answer set. These representations derived from the A-SIG allow a new form of reasoning, *i.e.*, one that combines visual understanding and structure reasoning. As shown in [32, 33, 34] and our experiments in Section 6, incorporating structure into RPM problem solving could result in further performance improvement across different models.

4.3. Human Performance Analysis

Another missing point in the previous work [3] is the evaluation of human performance. To fill in the missing piece, we recruit human subjects consisting of college students from a subject pool maintained by the Department of Psychology to test their performance on a subset of representative samples in the dataset. In the experiments, human subjects were familiarized by solving problems with only

one non-Constant rule in a fixed configuration. After the familiarization, subjects were asked to answer RPM problems with complex rule combinations, and their answers were recorded. Note that we deliberately included all figure configurations to measure generalization in the human performance and only “easily perceptible” examples were used in case certain subjects might have impaired perception. The results are reported in Table 2. The notable performance gap calls for further research into this problem. See Section 6 for detailed analysis and comparisons with vision models.

4.4. Heuristics-based Solver using Searching

We also find that the RPM could be essentially turned into a searching problem, given the symbolic representation of images and the access to rule operations as in [32, 33, 34]. Under such a setting, we could treat this problem as constraint satisfaction and develop a heuristics-based solver. The solver checks the number of satisfied constraints in each candidate answer and selects one with the highest score, resulting in perfect performance. Results are reported in Table 2. The optimality of the heuristic-based solver also verifies the well-formedness of RAVEN in the sense that there exists only one candidate that satisfies all constraints.

5. Dynamic Residual Tree for RPM

The image space of RPM is inherently structured and could be described using a symbolic language, as shown in [7, 32, 33, 34, 47]. To capture this characteristic and further improve the model performance on RPM, we propose a simple tree-structure neural module called Dynamic Residual Tree (DRT) that operates on the joint space of image understanding and structure reasoning. An example of DRT is shown in Figure 5.

In the DRT, given a sentence S sampled from the A-SIG, usually represented as a serialized n -ary tree, we could first recover the tree structure. Note that the tree is **dynamically** generated following the sentence S , and each node in the tree comes with a label. With a structured tree representation ready, we could now consider assigning a neural computation operator to each tree node, similar to Tree-LSTM [53]. To further simplify computation, we replace the LSTM cell [15] with a ReLU-activated [41] fully-connected layer f . In this way, nodes with a single child (leaf nodes or OR-production nodes) update the input fea-

Figure 5. An example computation graph of DRT. (a) Given the serialized n -ary tree representation (pre-order traversal with / denoting end-of-branch), (b) a tree-structured computation graph is dynamically built. The input features are wired from bottom-up following the tree structure. The final output is the sum with the input, forming a residual module.

tures by

$$I = \text{ReLU}(f([I, w_n])), \quad (1)$$

where $[\cdot, \cdot]$ is the concatenation operation, I denotes the input features, and w_n the distributed representations of the node's label [40, 45]. Nodes with multiple children (AND-production nodes) update input features by

$$I = \text{ReLU} \left(f \left(\bigcup_c I_c, w_n \right) \right), \quad (2)$$

where I_c denotes the features from its child c .

In summary, features from the lower layers are fed into the leaf nodes of DRT, gradually updated by Equation 1 and Equation 2 from bottom-up following the tree structure, and output to higher-level layers.

Inspired by [14], we make DRT a **residual** module by adding the input and output of DRT together, hence the name Dynamic Residual Tree (DRT)

$$I = \text{DRT}(I, S) + I. \quad (3)$$

6. Experiments

6.1. Computer Vision Models

We adopt several representative models suitable for RPM and test their performances on RAVEN [3, 14, 27, 57]. In summary, we test a simple sequential learning model (LSTM), a CNN backbone with an MLP head (CNN), a ResNet-based [14] image classifier (ResNet), the recent relational WReN [3], and all these models augmented with the proposed DRT.

LSTM The partially sequential nature of the RPM problem inspires us to borrow the power of sequential learning. Similar to ConvLSTM [57], we feed each image feature extracted by a CNN into an LSTM network sequentially and pass the last hidden feature into a two-layer MLP to predict the final answer. In the DRT-augmented LSTM, *i.e.*, LSTM-DRT, we feed features of each image to a shared DRT before the final LSTM.

CNN We test a neural network model used in Hoshen and Werman [17]. In this model, a four-layer CNN for image feature extraction is connected to a two-layer MLP with a softmax layer to classify the answer. The CNN is interleaved with batch normalization [20] and ReLU non-linearity [41]. Random dropout [51] is applied at the penultimate layer of MLP. In CNN-DRT, image features are passed to DRT before MLP.

ResNet Due to its surprising effectiveness in image feature extraction, we replace the feature extraction backbone in CNN with a ResNet [14] in this model. We use a publicly available ResNet implementation, and the model is randomly initialized without pre-training. After testing several ResNet variants, we choose ResNet-18 for its good performance. The DRT extension and the training strategy are similar to those used in the CNN model.

WReN We follow the original paper [3] in implementing the WReN. In this model, we first extract image features by a CNN. Each answer feature is then composed with each context image feature to form a set of ordered pairs. The order pairs are further fed to an MLP and summed. Finally, a softmax layer takes features from each candidate answer and makes a prediction. In WReN-DRT, we apply DRT on the extracted image features before the relational module.

For all DRT extensions, nodes in the same level share parameters and the representations for nodes' labels are fixed after initialization from corresponding 300-dimension GloVe vectors [45]. Sentences used for assembling DRT could be either retrieved or learned by an encoder-decoder. Here we report results using retrieval.

6.2. Experimental Setup

We split the RAVEN dataset into three parts, 6 folds for training, 2 folds for validation, and 2 folds for testing. We tune hyper-parameters on the validation set and report the model accuracy on the test set. For loss design, we treat the problem as a classification task and train all models with the cross-entropy loss. All the models are implemented in PyTorch [44] and trained with ADAM [25] before early stopping or a maximum number of epochs is reached.

6.3. Performance Analysis

Table 2 shows the testing accuracy of each model trained on RAVEN, against the human performance and the heuristics-based solver. Neither human subjects nor the solver experiences an intensive training session, and the solver has access to the rule operations and searches the answer based on a symbolic representation of the problem. In contrast, all the computer vision models go over an extensive training session, but only on the training set.

In general, human subjects produce better testing accuracy on problems with simple figure configurations such as **Center**, while human performance reasonably deteriorates on problem instances with more objects such as

Table 2. Testing accuracy of each model against human subjects and the solver. Acc denotes the mean accuracy of each model, while other columns show model accuracy on different figure configurations. L- R denotes Left- Right, U- D denotes Up- Down, O- IC denotes Out- InCenter, and O- IG denotes Out- InGrid. Note that the perfect solver has access to rule operations and searches on the symbolic problem representation.

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

2x2Grid and **3x3Grid**. Two interesting observations:

1. For figure configurations with multiple components, although each component in **Left- Right**, **Up- Down**, and **Out- InCenter** has only one object, making the reasoning similar to **Center** except that the two components are independent, human subjects become less accurate in selecting the correct answer.
2. Even if **Up- Down** could be regarded as a simple transpose of **Left- Right**, there exists some notable difference. Such effect is also implied by the “inversion effects” in cognition; for instance, inversion disrupts face perception, particularly sensitivity to spatial relations [8, 29].

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

We observe a consistent performance improvement across different models after incorporating DRT, suggesting the effectiveness of the structure information in this visual reasoning problem. While the performance boost is only marginal in LSTM and WReN, we notice a marked accuracy increase in the CNN- and ResNet-based models (6.63% and 16.58% relative increase respectively). How-

ever, the performance gap between artificial vision systems and humans are still significant (up to 37% in **2x2Grid**), calling for further research to bridge the gap.

6.4 Effects of Auxiliary Training

Barrett *et al.* [3] mentioned that training WReN with a fine-tuned auxiliary task could further give the model a 10% performance improvement. We also test the influence of auxiliary training on RAVEN. First, we test the effects of an auxiliary task to classify the rules and attributes on WReN and our best performing model ResNet+DRT. The setting is similar to [3], where we perform an OR operation on a set of multi-hot vectors describing the rules and the attributes they apply to. The model is then tasked to both correctly find the answer and classify the rule set with its governing attributes. The final loss becomes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \lambda \mathcal{L}_{\text{rule}}, \quad (4)$$

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

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

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \lambda \mathcal{L}_{\text{struct}}, \quad (5)$$

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

6.5. Test on Generalization

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

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

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

Table 3. Generalization test. The model is trained on **Center** and tested on three other configurations.

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

Table 4. Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

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

Table 5. Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

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

Table 3, 4 and 5 show the result of our model generalization test. We observe:

- The model dedicated to a single figure configuration does not achieve better test accuracy than one trained on all configurations together. This effect justifies the importance of the diversity of RAVEN, showing that increasing the number of figure configurations could actually improve the model performance.
- Table 3 also implies that a certain level of compositional reasoning, though weak, exists in the model, as the three other configurations could be regarded as a multi-component composition of **Center**.

- In Table 4, we observe no major differences in terms of test accuracy. This suggests that the model could successfully transfer the knowledge learned in a scenario to a very similar counterpart, when one configuration is the transpose of another.
- From Table 5, we notice that the model trained on **3x3Grid** could generalize to **2x2Grid** with only minor difference from the one dedicated to **2x2Grid**. This could be attributed to the fact that in the **3x3Grid** configuration, there could be instances with object distribution similar to that in **2x2Grid**, but not vice versa.

7. Conclusion

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

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

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

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

Acknowledgement: The authors thank Yuxin Chi and Prof. Ying Nian Wu and Prof. Hongjing Lu at UCLA Statistics Department for helpful discussions. The work reported herein was supported by DARPA XAI grant N66001-17-2-4029, ONR MURI grant N00014-16-1-2007, ARO grant W911NF-18-1-0296, and a NVIDIA GPU donation.

References

- [1] S. Aditya, Y. Yang, and C. Baral. Explicit reasoning over end-to-end neural architectures for visual question answering. *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018. **3**
- [2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015. **1**
- [3] D. Barrett, F. Hill, A. Santoro, A. Morcos, and T. Lillicrap. Measuring abstract reasoning in neural networks. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 511–520, 2018. **2, 3, 4, 5, 6, 7**
- [4] Y. Bisk, K. J. Shih, Y. Choi, and D. Marcu. Learning interpretable spatial operations in a rich 3d blocks world. *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018. **3**
- [5] F. W. Campbell and J. Robson. Application of fourier analysis to the visibility of gratings. *The Journal of physiology*, 197(3):551–566, 1968. **1**
- [6] Q. Cao, X. Liang, B. Li, G. Li, and L. Lin. Visual question reasoning on general dependency tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. **3**
- [7] P. A. Carpenter, M. A. Just, and P. Shell. What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. *Psychological review*, 97(3):404, 1990. **1, 2, 3, 4, 5**
- [8] K. Crookes and E. McKone. Early maturity of face recognition: No childhood development of holistic processing, novel face encoding, or face-space. *Cognition*, 111(2):219–247, 2009. **7**
- [9] R. E. Snow, P. Kyllonen, and B. Marshalek. The topography of ability and learning correlations. *Advances in the psychology of human intelligence*, pages 47–103, 1984. **2**
- [10] T. Evans. *A Heuristic Program to Solve Geometric Analogy Problems*. PhD thesis, MIT, 1962. **2**
- [11] T. G. Evans. A heuristic program to solve geometric-analogy problems. In *Proceedings of the April 21-23, 1964, spring joint computer conference*, 1964. **2**
- [12] K. S. Fu. *Syntactic methods in pattern recognition*, volume 112. Elsevier, 1974. **2**
- [13] C.-e. Guo, S.-C. Zhu, and Y. N. Wu. Primal sketch: Integrating structure and texture. *Computer Vision and Image Understanding (CVIU)*, 106(1):5–19, 2007. **1**
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. **6**
- [15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. **5**
- [16] K. J. Holyoak, K. J. Holyoak, and P. Thagard. *Mental leaps: Analogy in creative thought*. MIT press, 1996. **1**
- [17] D. Hoshen and M. Werman. Iq of neural networks. *arXiv preprint arXiv:1710.01692*, 2017. **3, 6**
- [18] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. **3**
- [19] D. A. Hudson and C. D. Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018. **3**
- [20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015. **6**
- [21] S. M. Jaeggi, M. Buschkuhl, J. Jonides, and W. J. Perrig. Improving fluid intelligence with training on working memory. *Proceedings of the National Academy of Sciences*, 105(19):6829–6833, 2008. **2**
- [22] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. **1, 2, 3**
- [23] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. **3**
- [24] G. Kanizsa and G. Kanizsa. *Organization in vision: Essays on Gestalt perception*, volume 49. Praeger New York, 1979. **1**
- [25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014. **6**
- [26] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. **2**
- [27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2012. **6**
- [28] M. Kunda, K. McGregor, and A. K. Goel. A computational model for solving problems from the raven’s

- progressive matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, 22:47–66, 2013. 2
- [29] R. Le Grand, C. J. Mondloch, D. Maurer, and H. P. Brent. Neuroperception: Early visual experience and face processing. *Nature*, 410(6831):890, 2001. 7
- [30] L. Lin, T. Wu, J. Porway, and Z. Xu. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, 42(7):1297–1307, 2009. 2
- [31] D. R. Little, S. Lewandowsky, and T. L. Griffiths. A bayesian model of rule induction in raven’s progressive matrices. In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2012. 3
- [32] A. Lovett and K. Forbus. Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124(1):60, 2017. 3, 5
- [33] A. Lovett, K. Forbus, and J. Usher. A structure-mapping model of raven’s progressive matrices. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2010. 3, 5
- [34] A. Lovett, E. Tomai, K. Forbus, and J. Usher. Solving geometric analogy problems through two-stage analogical mapping. *Cognitive science*, 33(7):1192–1231, 2009. 3, 5
- [35] D. Marr. *Vision: A computational investigation into*. WH Freeman, 1982. 1
- [36] D. Mascharka, P. Tran, R. Soklaski, and A. Majumdar. Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3
- [37] K. McGregor and A. K. Goel. Confident reasoning on raven’s progressive matrices tests. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 380–386, 2014. 3
- [38] K. McGregor, M. Kunda, and A. Goel. Fractals and ravens. *Artificial Intelligence*, 215:1–23, 2014. 3
- [39] C. S. Mekik, R. Sun, and D. Y. Dai. Similarity-based reasoning, raven’s matrices, and general intelligence. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1576–1582, 2018. 3
- [40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013. 6
- [41] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010. 5, 6
- [42] A. Newell. You can’t play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. Chase, editor, *Visual Information Processing: Proceedings of the Eighth Annual Carnegie Symposium on Cognition*. Academic Press, 1973. 2
- [43] S. Park and S.-C. Zhu. Attributed grammars for joint estimation of human attributes, part and pose. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 2
- [44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6
- [45] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 6
- [46] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3
- [47] J. C. e. a. Raven. Raven’s progressive matrices. *Western Psychological Services*, 1938. 2, 5
- [48] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015. 1
- [49] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017. 3
- [50] S. Shegheva and A. K. Goel. The structural affinity method for solving the raven’s progressive matrices test for intelligence. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3
- [51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 2014. 6
- [52] C. Strannegård, S. Cirillo, and V. Ström. An anthropomorphic method for progressive matrix problems. *Cognitive Systems Research*, 22:35–46, 2013. 2
- [53] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015. 2, 5
- [54] L. L. Thurstone and T. G. Thurstone. Factorial studies of intelligence. *Psychometric monographs*, 1941. 2

- [55] K. Wang and Z. Su. Automatic generation of raven’s progressive matrices. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2015. 2, 3, 4, 5
- [56] T.-F. Wu, G.-S. Xia, and S.-C. Zhu. Compositional boosting for computing hierarchical image structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 2
- [57] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation now-casting. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015. 6
- [58] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *arXiv preprint arXiv:1810.02338*, 2018. 1, 3
- [59] C. Zhu, Y. Zhao, S. Huang, K. Tu, and Y. Ma. Structured attentions for visual question answering. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017. 3
- [60] J. Zhu, T. Wu, S.-C. Zhu, X. Yang, and W. Zhang. A reconfigurable tangram model for scene representation and categorization. *IEEE Transactions on Image Processing*, 25(1):150–166, 2016. 2
- [61] S.-C. Zhu, D. Mumford, et al. A stochastic grammar of images. *Foundations and Trends^R in Computer Graphics and Vision*, 2(4):259–362, 2007. 2
- [62] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1