Hierarchical Rule Induction Network for Abstract Visual Reasoning

Sheng Hu*, Yuqing Ma*, Xianglong Liu†, Yanlu Wei, Shihao Bai
State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
husheng_7@163.com, {mayuqing,xlliu}@nlsde.buaa.edu.cn, {weiyanlu,16061167}@buaa.edu.cn

Abstract

Abstract reasoning refers to the ability to analyze information, discover rules at an intangible level, and solve problems in innovative ways. Raven's Progressive Matrices (RPM) test is typically used to examine the capability of abstract reasoning. In the test, the subject is asked to identify the correct choice from the answer set to fill the missing panel at the bottom right of RPM (e.g., *a 3×3 matrix), following the underlying rules inside the matrix. Recent studies, taking advantage of Convolutional Neural Networks (CNNs), have achieved encouraging progress to accomplish the RPM test problems. Unfortunately, simply relying on the relation extraction at the matrix level, they fail to recognize the complex attribute patterns inside or across rows/columns of RPM. To address this problem, in this paper we propose a Hierarchical Rule Induction Network (HriNet), by intimating human induction strategies. HriNet extracts multiple granularity rule embeddings at different levels and integrates them through a gated embedding fusion module. We further introduce a rule similarity metric based on the embeddings, so that HriNet can not only be trained using a tuplet loss but also infer the best answer according to the similarity score. To comprehensively evaluate HriNet, we first fix the defects contained in the very recent RAVEN dataset and generate a new one named Balanced-RAVEN. Then extensive experiments are conducted on the large-scale dataset PGM and our Balanced-RAVEN, the results of which show that HriNet outperforms the state-of-the-art models by a large margin.*

1. Introduction

Abstract reasoning, also known as inductive reasoning, refers to the ability to analyze information, discover rules at an intangible level, and solve problems in innovative ways. This type of reasoning, as the foundation for human intelligence, helps human understand the world. It has been generally regarded and pursued as a critical component to the development of artificial intelligence during the past decades, and has attracted increasing attention in recent years. Raven's Progressive Matrices (RPM) test [3, 11, 23, 29] is one of the highly accepted and well-studied tools to examine the ability of abstract reasoning, which is believed as a good estimate of the real intelligence [2]. An illustration of RPM is shown in Figure 1, where usually the test-taker is presented with a 3×3 matrix with the bottom right panel left blank. The goal is to choose one image from an answer set of eight candidates to complete the matrix correctly, namely satisfying the underlying rules in the matrix. Subjects accomplish this by looking into the first two rows/columns and inducing the dominant rules which govern the attributes in those panels. The obtained rules can then be applied to the last row/column to figure out which answer belongs to the blank panel.

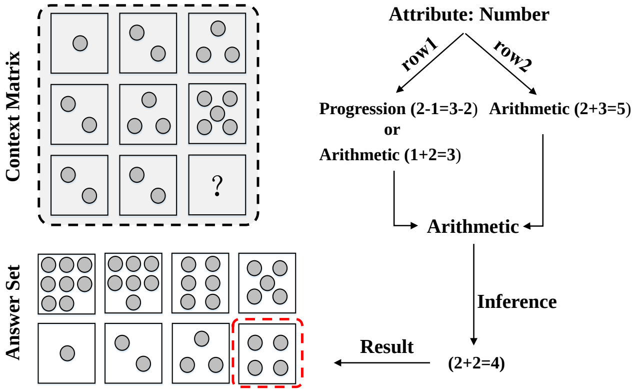

Figure 1. An example of RPM question and its solution. The underlying rule on the number of circles could be *Progression* (2-1=3-2) or *Arithmetic* (1+2=3) along row 1, and *Arithmetic* (2+3=5) along row 2. Therefore the dominant rule is *Arithmetic*. Apply it to the third row to figure out the answer (2+2=4). Besides, no viable rule can be found along the columns.

Computational models for RPM in the cognitive science community access symbolic representations of the images [14, 15, 16]. Recently there has been some success with end-to-end learning methods trying to accomplish abstract reasoning on RPM test [8, 1, 34, 35], inspired by the progress of computer vision tasks [5, 10, 33, 26, 30]. Bar-

*These authors contributed equally to this work.
†Corresponding authors.

rett *et al*. [1] proposed the Procedurally Generated Matrices (PGM) dataset constructed with relation-object-attribute tuples, which was automatically generated by a computer program. They also designed the Wild Relational Network (WReN) to learn a probability score for each multiple-choice panel infilled. Zhang *et al*. [34] built another large-scale dataset Relational and Analogical Visual rEasoNing (RAVEN) with structure annotations. RAVEN contains diverse rule instantiations, structures, and figure configurations, making it more comprehensive compared with the PGM dataset. Unfortunately, previous deep learning-based models simply relied on the relation extraction at the matrix level, and thus failed to recognize the complex attribute patterns inside or across rows/columns of RPM.

In this paper, we develop a novel architecture called Hierarchical Rule Induction Network (HriNet) inspired by human induction strategies. HriNet induces the underlying rules from the two given rows/columns, by extracting multiple granularity rule embeddings at different levels, including cell-wise, individual-wise, and ecological embeddings. The cell-wise hierarchy focuses on the attributes inside each panel, such as size, type, *etc*. The individual-wise hierarchy further takes the relationships inside each row/column into consideration. The ecological hierarchy comprehensively handles the correlations among all the panels within the two given sequences. These hierarchical embeddings are also fused in a hierarchical way using gate functions, to induce the shared rule embeddings between the two inputs. In order to determine the fitness of the candidate answer according to the extracted rule, we further introduce a rule similarity metric, based on which HriNet can not only be well trained using a tuplet loss but also quickly infer the best answer.

To fairly evaluate the capability of abstract reasoning, it is fundamental to build an unbiased RPM-style dataset. However, by taking a close look at the recently published RAVEN dataset [34], we find that there exist severe defects (or obvious patterns) among the answer set, where the correct one can be easily found without considering the context panels. A neural network trained with only the eight multiple-choice panels as input can surprisingly achieve 90.1% test accuracy. Such inappropriate setting has caused misleading results in the recent research [34, 35]. To fix the defects of RAVEN, we propose a new way of generating the answer set and name the unbiased dataset Balanced-RAVEN. Finally, we extensively evaluate our HriNet on the popular PGM dataset and our Balanced-RAVEN. The experimental results show that HriNet outperforms state-of-the-art methods by a large margin, *e.g*. 63.9% accuracy compared to the second best 44.3% on Balanced-RAVEN.

2. Related work

Computational models for solving RPM in the cognitive science community was based on an oversimplified assumption that computer programs had access to symbolic inputs of images and the operations of rules [2, 14, 15, 16]. Another research branch [13, 18, 19, 20, 25] explored RPM through measuring the similarity between images. Hoshen *et al*. [8] first trained a CNN-based model, trying to resolve RPM problems from raw pixels on a simplified RPM-style dataset. Wang and Su [31] proposed an automatic method to generate RPM questions using a computer program. Barrett *et al*. [1] borrowed the insight from [31] and introduced the Procedurally Generating Matrices (PGM) dataset. They also designed the Wild Relational Network (WReN) for RPM which took the pair-wise relationships among panels into consideration. Hill *et al*. [6] proposed a training strategy to learn analogies by contrasting abstract relational structure (LABC). Zhang *et al*. [34] adopted Attributed Stochastic Image Grammar (A-SIG) [4, 12, 22, 32, 36, 37] as the hierarchical image syntax to represent RPM questions and introduced another RPM-style dataset named Relational and Analogical Visual rEasoNing (RAVEN). Based on the rich annotations provided by A-SIG for each problem instance, they further designed a plug and play module called Dynamic Residual Tree (DRT), trying to improve the performance on RPM using the annotations of image structure. However, there are some unexpected defects contained in the RAVEN dataset which we will discuss in details in Section 4.1.2. They ulteriorly discussed the order-invariant characteristic of RPM and proposed CoPINet in [35]. However, the reported results conducted on the biased RAVEN dataset can not be used as reference.

3. Our approach

In this section, we first give a formal definition of the abstract reasoning task on the RPM test. Then we introduce the motivation from the human reasoning strategies, and subsequently present our Hierarchical Rule Induction Network (HriNet) for this task. Finally, we demonstrate the learning and inference process of the proposed model.

3.1. Preliminary

For a common RPM problem, usually a 3×3 matrix $\mathbf{M}^-$ is given, with bottom right context panel left blank. Ω denotes the answer set with N multiple-choice panels, where typically N=8. The dominant rules governing the features inside the matrix could be inducted from the first two intact rows/columns. The goal is to select a multiple-choice panel $\omega \in \Omega$ to complete the context matrix $\mathbf{M}^-$, maintaining the dominant rule inside of the context matrix.

We define the completed matrix with a multiple-choice panel ω infilled as $\mathbf{M}$, where $\mathbf{M}_i$ is denoted as the i-th row, and $\mathbf{m}_{ij}$ indicates the panel in i-th row and j-th column. Intuitively, $\mathbf{M}$ is almost the same as $\mathbf{M}^-$, except for $\mathbf{m}_{33} = \omega$ while the corresponding element missing in $\mathbf{M}^-$. In fact, whether rules exist in rows or columns is uncertain. Therefore, our framework induces both the row-wise rule representation and the column-wise representation in the same way. In order to simplify the notation, we only take the induction of the row-wise rule representation as example.

3.2. The reasoning framework

We develop a novel abstract reasoning architecture named Hierarchical Rule Induction Network (HriNet), inspired by hierarchical induction strategies of human. As shown in Figure 1, given a Raven's Progressive Matrix, human strategies can be simplified into five key steps:

S1: look into each panel, including context panel and multiple-choice panel, to recognize the basic attributes of the graphical elements, *e.g.*, type, size, color, position.

S2: compare panels in the same row to figure out the plausible rules inside it.

S3: compare panels in two rows to figure out the shared relationships between the two rows.

S4: scan the first two rows and induce the dominant rules, integrating hierarchical information from previous 3 step.

S5: fill each multiple-choice panel in the matrix, infer the rules with hierarchical information as S4 did according to each currently completed matrix, and determine the correct answer which adheres to the dominant rule.

Given two input rows $\mathbf{M}_i, \mathbf{M}_j$, the proposed framework adopts similar strategies as humans do. It embeds the input into the multiple granularity embeddings at different levels using a hierarchical rule embedding module $\mathbb{E}$. Inspired by human reasoning strategy and the general information processing mechanism in the biological organization [21], $\mathbb{E}$ consists of three hierarchies, namely cell-wise network $\mathbb{E}_{\text{cell}}$, individual-wise network $\mathbb{E}_{\text{ind}}$, and ecological network $\mathbb{E}_{\text{eco}}$, which respectively look into the matrix from different hierarchies, focusing on the attribute and pattern discovery from cell-wise hierarchy as S1, individual-wise hierarchy as S2, and ecological hierarchy as S3.

With the multiple granularity rule embeddings, the gated embedding fusion module $\mathbb{G}$ will integrate these hierarchical features and induce the final rule embedding $\mathbf{r}_{ij}^{(3)}$ of the two input sequences $\mathbf{M}_i$ and $\mathbf{M}_j$. The embedding representation of the rules preserves the semantic distances among rules, namely keep that of similar rules close and dissimilar rules far in the embedding space. Therefore, we further introduce a rule similarity metric $\mathcal{D}$ to estimate the similarity between the rule representations. As a result, the correct answer can be predicted by choosing the multiple-choice panel within the shortest distance to the dominant rule generated by the first two rows in the matrix, like S4 and S5 in the human reasoning process.

3.3. Hierarchical Rule Induction Network

Now we introduce the carefully designed hierarchical modules in our framework in details.

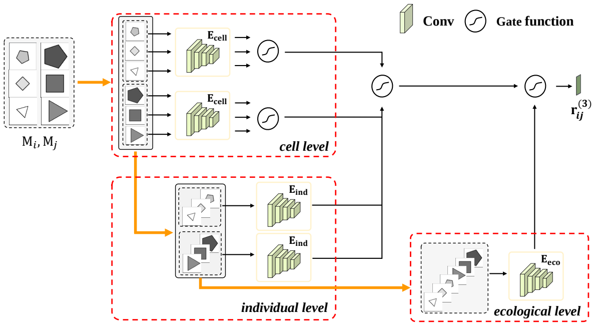
Conv
Gate function
$\mathrm{M}_i, \mathrm{M}_j$
$\mathbf{E}_{\text{cell}}$
$\mathbf{E}_{\text{cell}}$
cell level
$\mathbf{r}_{ij}^{(3)}$
$\mathbf{E}_{\text{ind}}$
$\mathbf{E}_{\text{ind}}$
individual level
$\mathbf{E}_{\text{eco}}$
ecological level

Figure 2. The architecture of HriNet, consisting of a hierarchical rule embedding module and a gated embedding fusion module. Given two row sequences as input, it outputs the rule embedding.

3.3.1 Hierarchical rule embedding

As we all know, organization of behaviour into a nested hierarchy of tasks is characteristic of purposive cognition in humans. The prevalent Convolution Neural Network inspired by the human visual system, is a hierarchical model itself, with the projection from each layer showing the hierarchical nature of features. The bottom layers extract low-level features, such as texture, edge, *etc*., while the top layers abstract high-level semantic information from the low-level information transmitted from the bottom layers.

However, without specifying information from different levels, it is hard for CNN to figure out different hierarchies, and thus fail to obtain robust and representative features. Therefore, it would be better to feed the input of different hierarchies explicitly and extract rule representations from different granularity with artificial guidance. Motivated by that, we deploy a hierarchical rule embedding module, consisting of cell-wise hierarchy, individual-wise hierarchy, and ecological hierarchy.

Cell-wise hierarchy The network of the cell-wise hierarchy $\mathbb{E}_{\text{cell}}$ takes each panel as input and recognize the attributes of inside graphical elements. It handles each panel independently without considering the difference or correlations among panels inside the matrix. Therefore, it observes the information from the most detailed perspective. We obtain the cell-wise rule representation for each input panel:

$$\mathbf{x}_{ij} = \mathbb{E}_{\text{cell}}(\mathbf{m}_{ij}). \tag{1}$$

Individual-wise hierarchy Moreover, the network of individual hierarchy takes each row as input. It begins to take the correlations among panels of the same row into consideration, and encode the entire row with a compact embedding, rather than simply combining each panel. In this way, the rule embedding process for each panel is coupled and interacts with each other. Intuitively, each row may contain multiple rules, such as color, number, *etc*. In this hierarchy, the framework extracts intermediate rule embedding for each row individually, which still ignores the comprehensive information from the matrix perspective, especially the correlations across rows. The individual-wise rule embedding $\mathbf{y}_i$ is denoted as:

$$\mathbf{y}_i = \mathbb{E}_{\text{ind}}(\mathbf{M}_i). \tag{2}$$

Ecological hierarchy Furthermore, the network of the ecological hierarchy takes the two rows together as input and jointly learns the rule patterns underlying the two rows. As we mentioned before, in the individual hierarchy, the framework extracts intermediate rule embedding for each row, without considering the interaction between two rows. The rule that exists in one row may not lie in another. Therefore, to obtain the shared rule patterns between the two rows, it is essential to put these two rows together and jointly learn the features from an ecological level. Thus the shared rule embedding is obtained as follows:

$$\mathbf{z}_{ij} = \mathbb{E}_{\text{eco}}([\mathbf{M}_i, \mathbf{M}_j]), \tag{3}$$

where $[\cdot, \cdot]$ denotes the concatenating operation.

3.3.2 Gated embedding fusion

Since the rule embeddings at different levels focus on different attributes or patterns, to generate one discriminative representation for the rule, we should aggregate the multiple granularity embeddings. However, due to the requirement that the aggregation should preserve the order of cell-wise rule embeddings and be invariant to the order of the individual-wise ones, it is impracticable to receive all the rule embeddings simultaneously relying on a single fully connected network. Therefore we propose a hierarchical rule embedding learning method named gated embedding fusion module, which is responsible for hierarchically and gradually aggregating the multiple granularity embeddings.

Specifically, we define a gate function φ to fuse the rule embeddings from different hierarchies. It concatenates all the inputs and encodes into a single embedding using fully connected layers. The gate function is similar to the attention mechanism, which detects and concentrates on the useful features according to the task. Even for the same attribute, they may focus on different facets. More details could be found in supplementary materials. Based on the gate function, our gated embedding fusion module could regulate the flow of rule embeddings into the framework and make the utmost of their complementary information.

At the cell level, after obtaining cell-wise rule embeddings for panels in each row $\mathbf{M}_i$, the module aggregates them to infer a row-wise rule embedding $\mathbf{r}_i^{(1)}$:

$$\mathbf{r}_i^{(1)} = \varphi_1(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{x}_{i3}), \tag{4}$$

Similarly, we obtain $\mathbf{r}_j^{(1)}$ for the j-th row $\mathbf{M}_j$. The fused embedding integrates different types of information in the panels such as type and size.

At the individual level, intuitively both $\mathbf{r}_i^{(1)}$ and $\mathbf{y}_i$ are the row-wise embeddings corresponding to the i-th row, but convey the different granularity rule information. We further fuse them, and jointly mine the shared rules contained in the i-th and j-th row :

$$\mathbf{r}_{ij}^{(2)} = \varphi_2(\mathbf{r}_i^{(1)}, \mathbf{y}_i, \mathbf{r}_j^{(1)}, \mathbf{y}_j). \tag{5}$$

At the ecological level, similarly we can further combine hierarchically fused embedding $\mathbf{r}_{ij}^{(2)}$ and $\mathbf{z}_{ij}$ using the gate fusion function, abstracting the final rule embedding:

$$\mathbf{r}_{ij}^{(3)} = \varphi_3(\mathbf{r}_{ij}^{(2)}, \mathbf{z}_{ij}). \tag{6}$$

In practice, to make sure the framework is order-invariant to the input rows, we can simply exchange the concatenation order between the two input rows and average their rule embeddings. This invariance ensures that, the induced rule embedding respects the property of RPM and thus distills the representative information of the relations existing in the inputs.

On the whole, the HriNet can be formulated in its simplest form as follows:

$$\begin{aligned} \mathbf{r}_{ij}^{(3)} &= \text{HriNet}(\mathbf{M}_i, \mathbf{M}_j) \\ &= \mathbb{G}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j, \mathbf{z}_{ij}), \end{aligned} \tag{7}$$

where $\mathbf{r}_{ij}^{(3)}$ is the shared rule embedding of the $\mathbf{M}_i$ and $\mathbf{M}_j$. An illustration of HriNet is shown in Figure 2.

3.4. Learning and inference

With HriNet framework, the question turns to how we train the network, and apply it to infer the correct answer to RPM test. The key to address the question lies in the similarity measure between two rule embeddings, based on which we can define the loss function for HriNet training, and meanwhile determine the best choice during inference.

Similarity function We first introduce similarity function $\mathcal{D}$ to measure the closeness between two rules in the embedding space. There are a number of candidate functions:

1. *Cosine similarity:*

$$\mathcal{D}(\mathbf{r}, \mathbf{r}') = \frac{\mathbf{r}^{\mathrm{T}}\mathbf{r}'}{\parallel \mathbf{r} \parallel \parallel \mathbf{r}' \parallel},$$

2. *Euclidean similarity:*

$$\mathcal{D}(\mathbf{r}, \mathbf{r}') = - \parallel \mathbf{r} - \mathbf{r}' \parallel_2^2,$$

3. *Inner product similarity:*

$$\mathcal{D}(\mathbf{r}, \mathbf{r}') = \mathbf{r}^{\mathrm{T}}\mathbf{r}'.$$

In this paper, we simply adopt inner product similarity.

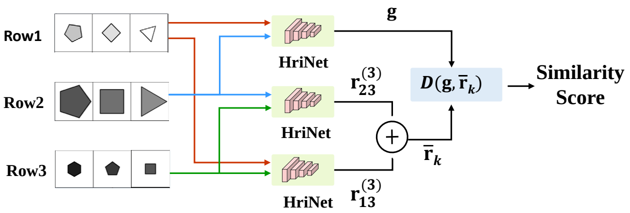

Figure 3. The similarity score for a candidate answer. A multiple-choice panel from the answer set is infilled in the blank panel (row 3), generating a rule embedding $\bar{\mathbf{r}}_k$ through HriNet. The similarity score for the candidate answer can be estimated based on $\bar{\mathbf{r}}_k$ and the dominant rule embedding $\mathbf{g}$ extracted from row 1 and 2.

Training For a given RPM problem, the first two rows $\mathbf{M}_1, \mathbf{M}_2$ are fed into our proposed HriNet and produce the shared rule embedding $\mathbf{g}$:

$$\mathbf{g} = \mathbf{r}_{12}^{(3)} = \text{HriNet}(\mathbf{M}_1, \mathbf{M}_2), \tag{8}$$

which represents the dominant pattern of the matrix.

Intuitively, the rule extracted from the first two rows can be treated as the reference rule, and we name it the dominant rule in the matrix. Subsequently, the correct answer can be found by checking whether its corresponding rule embedding is similar to the dominant rule. Specifically, given a multiple-choice panel $\omega_k \in \Omega$, where $k \in \{1, ..., N\}$, we denote $\bar{\mathbf{r}}_k$ as the new rule embedding inside $\mathbf{M}$ caused by the k-th multiple-choice panel:

$$\bar{\mathbf{r}}_k = \frac{1}{2}\left(\mathbf{r}_{13}^{(3)} + \mathbf{r}_{23}^{(3)}\right). \tag{9}$$

This procedure is illustrated in Figure 3. In practice, we generate the column-wise rule representation just as the row-wise one, and concatenate the two representations together as the final representation.

For the rule embedding $\bar{\mathbf{r}}^*$ generated by rows/columns infilled with correct answer, the desirable HriNet should enforce it to be more similar to the dominant rule $\mathbf{g}$, compared to the other rules $\bar{\mathbf{r}}_k$ corresponding to the wrong answers, where $\bar{\mathbf{r}}_k \neq \bar{\mathbf{r}}^*$. Subsequently, the generated rules of N candidates, alongside with the dominant rule, form a tuple containing N+1 elements. Based on the similarity function, the (N+1)-tuplet loss [27] can be defined for HriNet training:

$$\mathcal{L} = \log(1 + \sum_{k=1, \bar{\mathbf{r}}_k \neq \bar{\mathbf{r}}^*}^{N} \exp(\mathcal{D}(\mathbf{g}, \bar{\mathbf{r}}_k) - \mathcal{D}(\mathbf{g}, \bar{\mathbf{r}}^*))), \tag{10}$$

which means the HriNet can be trained in a fully end-to-end manner. The architecture of the HriNet (Figure 2) is well matched to the problem of abstract reasoning, because it leverages human strategies and explicitly generates the rules governing the matrix.

Inference Once the training of HriNet is finished, we could make the inference of the newly given RPM problem. Initially, the intact rows/columns of the RPM are fed into the framework to get the dominant rule $\mathbf{g}$. After that, each multiple-choice panel is filled to the blank position to complete the matrix, and the framework will generate the rule embeddings $\bar{\mathbf{r}}_k$ for all candidate answers, given the current completed matrix. We can accomplish the abstract reasoning by choosing the correct multiple-choice as follows:

$$k^* = \arg\max_k \mathcal{D}(\mathbf{g}, \bar{\mathbf{r}}_k). \tag{11}$$

Note that since we investigate each panel independently, the above inference framework promises that our model matches the nature of RPM that the answer should be invariant to the order of multiple-choice panels.

4. Experiments

4.1. Datasets

To comprehensively evaluate our model, we choose the recently proposed RAVEN [34] and PGM [1] datasets. Next, we first give a brief review of the two datasets, then we will demonstrate the defects of the original RAVEN and introduce an improved dataset named Balanced-RAVEN.

4.1.1 RAVEN and PGM datasets

RAVEN [34] It consists of 70,000 RPM questions, distributed in 7 different figure configurations (`Center`, `2x2Grid`, `3x3Grid`, `Out-InCenter`, `Out-InGrid`, `Left-Right`, and `Up-Down`). Panels are constructed with 5 attributes (`Number`, `Position`, `Type`, `Size`, and `Color`). Each attribute is governed by one of 4 rules (`Constant`, `Progression`, `Arithmetic`, and `Distribute Three`) and takes a value from a predefined set. Rules are applied only row-wise in RAVEN.

PGM [1] It contains 1.42M RPM questions. Rules in a matrix are composed with 1 to 4 relation-object-attribute tuples and can be applied along the rows or columns. For a fair comparison with the state-of-the-art abstract reasoning methods, we randomly sample 70,000 questions which is of the same size as RAVEN for experiments according to the underlying relations, making sure that it covers all 29 relations in this dataset. We denote the dataset as **PGM-70K**.

4.1.2 Balanced-RAVEN

After carefully examining the data in RAVEN, we find that there is unexpected bias among the eight multiple-choice panels. Each distractor in the answer set is generated by randomly modifying one attribute of the correct answer (see Figure 4(a)). As a consequence, the panel with the most common values for each attribute will be the correct answer.

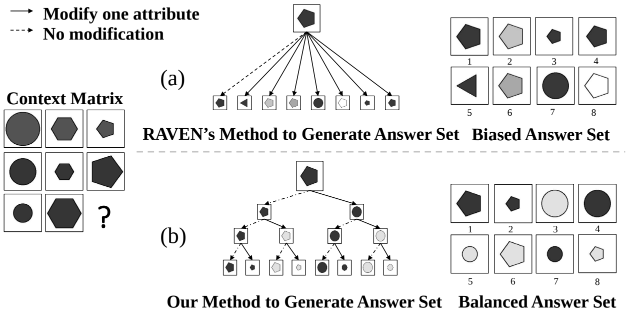

Figure 4. Comparison between RAVEN and Balanced-RAVEN.

This means that the correct answer can be found by simply scanning the answer set without considering the context images. An example is also shown on the right of Figure 4(a). Among the answer set, the most common `Color` and `Type` are black (No. 1, 3, 4, 5, and 7) and pentagon (No. 1, 2, 3, 4, 6, and 8). Besides, multiple-choice panel 1, 2, 5, 6, 7, and 8 are in the same `Size`. Therefore, multiple-choice panel 1 is the panel with the most common attribute values, which is indeed the correct answer to the RPM test.

More severely, as shown in Table 1, such underlying patterns can also be easily detected by a neural network. We simply train two models including a normal abstract reasoning model based on ResNet classifier (as detailed in Section 4.2) and a context-blind [1] ResNet model. The context-blind ResNet model is trained with only eight multiple-choice panels as input, without considering the context. It is very surprising that the context-blind model can get close (even slightly better) performance to the normal ResNet. It is worth noting that, here we adopt a simple data augmentation method that shuffles candidate images during training. This augmentation method is essential especially for models taking the whole answer set as input. Therefore, our result is much higher than the accuracy (53.43%) reported in [34].

Model	RAVEN	Balanced-RAVEN
ResNet	89.2%	40.3%
Context-blind ResNet	90.1%	12.5%

Table 1. Test on RAVEN and Balanced-RAVEN.

To fix the defects of RAVEN, we design an algorithm to generate the unbiased answer set, forming an improved dataset named Balanced-RAVEN. Figure 4(b) demonstrates the generating process using a tree structure. Each node indicates a multiple-choice panel, and the root of the tree structure is the correct answer. Different levels indicate different iterations, where nodes of this level are the candidate answers of current answer set. The generating process flows in a top-down manner. For each iteration, only one attribute will be modified. At each level, a node has two children nodes, where one node remains the same with the father node, the other changes the value of the attribute sampled for this iteration of the father node. Finally, at the bottom level, we could obtain the whole answer set. Algorithm 1 summarizes the key steps of the answer generating process.

Since the attribute modification is well balanced, no clue can be found to guess the answer only depending on the answer set. The right column in Table 1 shows that the performance of context-blind ResNet trained on Balanced-RAVEN is almost at a random guess level (12.5%), while the normal ResNet model further relying on the context can obtain much better performance. This observation proves that our improved dataset is more rigorous and fair for evaluating the capability of abstract reasoning.

Algorithm 1 Generating the Balanced-RAVEN

```
Input: the correct answer $\omega^*$
Initialize the answer set $\Omega = \{\omega^*\}$
Sample 3 attributes $a_1, a_2, a_3$ according to $\omega^*$
Sample new value $v_i$ for each $a_i$
for $i = 1$ to 3 do
    Initialize $\Gamma = \{\}$
    for each $w_k$ in the current answer set $\Omega$ do
        $\gamma \leftarrow$ modifying attribute $a_i$ of $\omega_k$ with $v_i$
        $\Gamma \leftarrow \Gamma \bigcup \{\gamma\}$
    end for
    $\Omega \leftarrow \Omega \bigcup \Gamma$
end for
Output: the answer set $\Omega$ ($|\Omega| = 2^3 = 8$)
```

4.2. Experimental setup

With PGM and Balanced-RAVEN, we first compare our method with several state-of-the-art models suited for RPM, including LSTM [7], ResNet-based [5] image classifier (ResNet), ResNet with DRT [34], Wild ResNet [1], WReN [1], and CoPINet [35]. Then we analyze the effects of different components in our HriNet.

We adopt the public implementations of LSTM, ResNet, and DRT in [34]. Eight context panels and eight multiple-choice panels are stacked and passed through the ResNet to output an 8-dimensional probability score. DRT is a plug and play module which could be deployed in any model. However, it cannot be applied to PGM-70K for the lack of structure annotations. Wild ResNet takes one multiple-choice panel, along with the eight context panels as input. It is designed to provide a score value for each multiple-choice panel, independent of the other multiple-choice panels. WReN, which takes the same input as Wild ResNet, applies a Relation Network [24] to obtain pairwise relationships among panels. We implement two versions of WReN, with its original 4-layer CNN or a ResNet-18 as the feature extractor. We haven't managed to implement CoPINet

Model	PGM-70K	Balanced-RAVEN	Center	2×2G	3×3G	O-IC	O-IG	L-R	U-D
LSTM	20.3	18.9	26.2	16.7	15.1	21.9	21.1	14.6	16.5
ResNet	21.7	40.3	44.7	29.3	27.9	46.2	35.8	51.2	47.4
ResNet+DRT		40.4	46.5	28.8	27.3	46.0	34.2	50.1	49.8
Wild ResNet	26.6	44.3	50.9	33.1	30.8	50.9	38.7	53.1	52.6
WReN	29.1	23.8	29.4	26.8	23.5	22.5	21.5	21.9	21.4
WReN (ResNet)	27.0	42.6	75.7	45.9	39.0	37.2	34.8	31.2	34.8
HriNet	**48.9**	**63.9**	**80.1**	**53.3**	**46.0**	**71.0**	**49.6**	**72.8**	**74.5**

Table 2. Test accuracy of different models. The left two columns show the average accuracy on PGM-70K and Balanced-RAVEN, while other columns show accuracy across seven figure configurations of Balanced-RAVEN. 2×2G, 3×3G, O-IC, O-IG, L-R and U-D denote `2x2Grid`, `3x3Grid`, `Out-InCenter`, `Out-InGrid`, `Left-Right` and `Up-Down`, respectively. The DRT module cannot be applied on PGM for the lack of structure annotations.

and test it on our Balanced-RAVEN, since it was published in the very recent past, and thus we only compare with its accuracy on PGM reported in [35].

For our HriNet, we adopt three ResNet-18 as the embedding networks for the three hierarchies, by only modifying the input channels. The gate fusion φ_1 and φ_2f are 2-layer fully connected networks, while φ_3 is a 4-layer fully connected network with dropout [28] of 0.5 applied on the last layer. We adopt stochastic gradient descent using ADAM [9] optimizer. The exponential decay rate parameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and the learning rate is 10^{-4}. On both datasets, 20-fold validation is performed to evaluate model performance, and the accuracy is averaged over 20 folds.

4.3. Comparisons with state-of-the-art methods

Table 2 lists the test accuracy of different models trained on PGM-70K and Balanced-RAVEN. From the table, it is obvious that our proposed HriNet outperforms other methods by a large margin on both datasets (18.7% and 19.6% accuracy increases respectively). Besides, we observe that models benefit from considering each multiple-choice panel independently, including the competitive Wild ResNet, WReN, and our HriNet. Such weight-sharing mechanism across panels can not only make a model invariant to the order of input multiple-choice panels, but also encourage to explore the underlying rules. Moreover, by comparing the results of two versions of WReN, we find that a deeper CNN backbone may improve the performance for abstract reasoning owing to the capability of extracting more complex patterns in the image. The very recent method CoPINet [35] achieved an accuracy of 32.39% when trained on a subset of PGM with 75,000 questions. Compared to CoPINet, our HriNet performs significantly better, when trained using a similar number of training data.

For more detailed comparison, Table 2 also reports the accuracy on seven figure configurations of Balanced-RAVEN. We can observe that accuracy on different configurations is not uniform, possibly due to the difficulty of configurations. But compare to other models, our HriNet consistently achieves the best performance on all the configurations, which proves that our model can work stably and robustly, even facing diverse conditions and complex rules. We will further interpret the reason in Section 4.5.

4.4. Ablation study

As aforementioned, our method mainly gains from the hierarchical architecture intimating human strategies. To validate this point, we study the effects of different hierarchies in abstract visual reasoning. Specifically, we set the rule embedding of certain hierarchy as a zero vector before gate function φ. Thus, the gate function regulates the flow of features into the gated embedding fusion module.

Table 3 lists the result of different choices of hierarchies. First, there is no doubt that our full model achieves the best performance compared to the other combinations, which indicates that all hierarchies contribute to our framework. Second, without considering the relationships among the panels, the performance of the cell-wise hierarchy is unsatisfactory, but still outperforms other state-of-the-art models (as shown in Table 2). That is to say, our strategy that explicitly induces the rule representation and then compares with the dominant rule is totally reasonable.

One more interesting observation is that a simple combination of two arbitrary hierarchies does not always achieves better performance than one hierarchy, due to the fact that they may focus on the same or mutually-exclusive attributes, since different hierarchies focus on different attributes. Figure 5(a) also supports this observation as well. We conduct experiments only utilizing single-hierarchy rule embeddings from $\mathbb{E}_{\text{cell}}, \mathbb{E}_{\text{ind}}, \mathbb{E}_{\text{eco}}$, on Balanced-RAVEN, with respect to three attributes (`Type`, `Size` and `Color`). As shown in Figure 5(a), $\mathbb{E}_{\text{cell}}$ has strong capacity to infer attributes `Type` and `Size`, but struggles to distinguish attribute `Color`. By contrast, $\mathbb{E}_{\text{ind}}$ and $\mathbb{E}_{\text{eco}}$ have modest ability to infer attributes `Type` and `Size`, and are efficient for attribute `Color`.

Model	PGM-70K	Balanced-RAVEN
$\mathbb{E}_{\text{cell}}$	34.1	36.7
$\mathbb{E}_{\text{ind}}$	42.2	48.7
$\mathbb{E}_{\text{eco}}$	41.9	51.6
$\mathbb{E}_{\text{cell}} + \mathbb{E}_{\text{ind}}$	44.8	57.8
$\mathbb{E}_{\text{cell}} + \mathbb{E}_{\text{eco}}$	40.6	52.9
$\mathbb{E}_{\text{ind}} + \mathbb{E}_{\text{eco}}$	42.0	57.0
$\mathbb{E}_{\text{cell}} + \mathbb{E}_{\text{ind}} + \mathbb{E}_{\text{eco}}$	**48.9**	**63.9**

Table 3. HriNet ($\mathbb{E}_{\text{cell}}$+$\mathbb{E}_{\text{ind}}$+$\mathbb{E}_{\text{eco}}$) and the results of eliminating different hierarchies.

4.5. The interpretability of rule embeddings

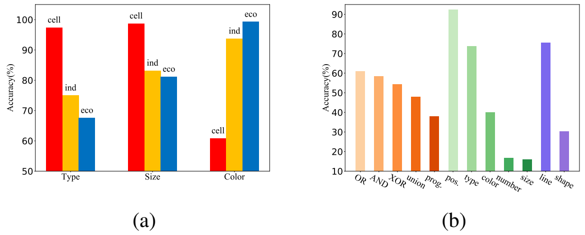

Figure 5. (a) Accuracy of single hierarchy with respect to the different attributes on Balanced-RAVEN. (b) Accuracy of HriNet with respect to the relation type, attribute type and object type on PGM-70K.

In the real RPM test, it is not clear whether the rule exists in rows or columns. However, it is important to check whether the proposed model can discover the knowledge without any guidance. Therefore, in this part we further interpret the reasoning behaviors of the model.

Rule induction for columns is normally left out when trained on Balanced-RAVEN, given the prior knowledge that rules are applied only row-wise. In order to test the ability of distinguishing whether the rules are applied along rows or columns, we train a HriNet model on Balanced-RAVEN which the induction for column rules is reintegrated into. As a result, there is only a bit drop in accuracy (from 63.9% to 59.6%). This indicates that our model can neglect the distraction brought by columns on its own.

Furthermore, since our model could induce the rule embeddings, we visualize these representations using the t-SNE [17] scatter. The scatter indicates whether our model can extract the semantic relations and encode this information in the rule embeddings. Naturally, we simply select certain PGM questions with only one dominant rule inside the matrix. Figure 6 respectively shows dominant rule representations of the matrix that are predicted right and wrong. Besides, to clearly understand our HriNet over different rule types, we also investigate the performance with respect to the relation type, attribute type and object type on PGM-70K dataset in Figure 5(b), where similar types (e.g., line and shape) are represented by similar colors.

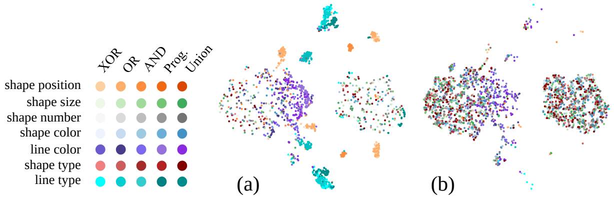

Figure 6. t-SNE scatter plots of the rule embeddings. (a) Dominant rule embeddings of correctly predicted questions. (b) Dominant rule embeddings of wrongly predicted questions.

From Figure 5(b), we get the consistent observation with that in Table 2, namely, the performance of our proposed model is imbalanced with respect to the difficulty of different configurations. Based on the observation, and further by comparing Figure 6(a) with Figure 6(b), we could further reach a conclusion that the performance of our model is positively related to the discrimination ability of the rule representations. For rules with good performance, such as `line type`, `shape position` and `line color`, they scatter closer as a dense cluster than those with poor performance. This further indicates that well-induced rule representations are helpful to find the correct answer to RPM test, and our HriNet owns strong capability of extracting discriminative rule embeddings.

5. Conclusion

In this paper, we proposed a novel Hierarchical Rule Induction Network for abstract visual reasoning task intimating human inducing strategies, which could extract multiple granularity rule embeddings at different level and integrate them through a gated embedding fusion module. A rule similarity metric was further introduced based on the embeddings, so that HriNet can not only be trained using a tuplet loss but also infer the best answer according to the similarity score. We also designed an algorithm to fix the defects of the very recent proposed dataset RAVEN, and generated a more rigorous dataset based on the algorithm. Extensive experiments conducted on PGM-70K dataset and our improved dataset Balanced-RAVEN proved that, our proposed framework could significantly outperform other state-of-the-art approaches. Moreover, we studied the effects of each component of our proposed model and evaluated the interpretability of our induced rule embeddings. Although existing learning based models show promising performance in abstract reasoning, they mainly rely on the abundance of training data, and struggle to transfer the reasoning ability to RPM questions with unseen rules. In the future, we will introduce meta-learning strategies into our framework to improve both the inducing and deducing abilities at the same time.

References

[1] 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)*, page 511520, 2018. 1, 2, 5, 6

[2] 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):404–431, 1990. 1, 2

[3] J. C. e.a. Raven. Ravens progressive matrices. *Western Psychological Services*, 1938. 1

[4] K. S. Fu. *Syntactic methods in pattern recognition*, volume 112. Elsevier, 1974. 2

[5] 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)*, pages 770–778, 2016. 1, 6

[6] F. Hill, A. Santoro, D. Barrett, A. Morcos, and T. Lillicrap. Learning to make analogies by contrasting abstract relational structure. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 6

[8] D. Hoshen and M. Werman. Iq of neural networks. *arXiv preprint arXiv:1710.01692*, 2017. 1, 2

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014. 7

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 1

[11] M. Kunda, K. McGreggor, and A. K. Goel. A computational model for solving problems from the ravens progressive matrices intelligence test using iconic visual representations. *Cognitive Systems Research*, 22:47–66, 2013. 1

[12] 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

[13] D. R. Little, S. Lewandowsky, and T. L. Griffiths. A bayesian model of rule induction in raven's progressive matrices. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, 2012. 2

[14] A. Lovett and K. Forbus. Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124(1):60–90. 1, 2

[15] 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*, volume 32, 2010. 1, 2

[16] A. Lovett, E. Tomai, K. Forbus, and J. Usher. Solving geometric analogy problems through two-stage analogical mapping. *Cognitive Science*, 33(7):1192–1231, 2010. 1, 2

[17] L. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[18] K. McGreggor and A. Goel. Confident reasoning on raven's progressive matrices tests. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014. 2

[19] K. McGreggor, M. Kunda, and A. Goel. Fractals and ravens. *Artificial Intelligence*, 215:1–23, 2014. 2

[20] C. S. Mekik, R. Sun, and D. Y. Dai. Similarity-based reasoning, raven's matrices, and general intelligence. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1576–1582, 2018. 2

[21] E Parent. The living systems theory of james grier miller. *Retrieved November*, 29:2007, 1996. 3

[22] S. Park and S.-C. Zhu. Attributed grammars for joint estimation of human attributes, part and pose. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2372–2380, 2015. 2

[23] J. Raven. The raven's progressive matrices: change and stability over culture and time. *Cognitive psychology*, 41(1):1–48, 2000. 1

[24] A. Santoro, D. Raposo, D. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4967–4976, 2017. 6

[25] S. Shegheva and A. Goel. The structural affinity method for solving the raven's progressive matrices test for intelligence. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 2

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[27] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1857–1865, 2016. 5

[28] 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*, 15(1):1929–1958, 2014. 7

[29] C. Strannegård, S. Cirillo, and V. Ström. An anthropomorphic method for progressive matrix problems. *Cognitive Systems Research*, 22:35–46, 2013. 1

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 1

[31] K. Wang and Z. Su. Automatic generation of ravens progressive matrices. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015. 2

[32] T. F. Wu, G. S. Xia, and S.-C. Zhu. Compositional boosting for computing hierarchical image structures. In *2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 2

[33] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833. Springer, 2014. 1

[34] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), pages 5317–5327, 2019. 1, 2, 5, 6

[35] C. Zhang, B. Jia, F. Gao, Y. Zhu, H. Lu, and S.-C. Zhu. Learning perceptual inference by contrasting. *Advances in Neural Information Processing Systems (NIPS)*, 2019. 1, 2, 6, 7

[36] 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, 2015. 2

[37] S.-C. Zhu, D. Mumford, and et. al. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2(4):259–362, 2007. 2