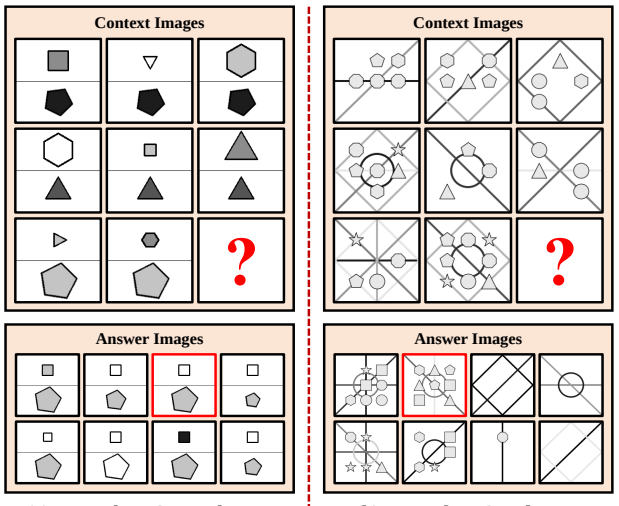
Neural Prediction Errors enable Analogical Visual Reasoning in Human Standard Intelligence Tests

Lingxiao Yang [1] **Hongzhi You** [2] **Zonglei Zhen** [3] **Da-Hui Wang** [3]
Xiaohong Wan [3] **Xiaohua Xie** [1] **Ru-Yuan Zhang** [4]

Abstract

Deep neural networks have long been criticized for lacking the ability to perform analogical visual reasoning. Here, we propose a neural network model to solve Raven's Progressive Matrices (RPM) — one of the standard intelligence tests in human psychology. Specifically, we design a reasoning block based on the well-known concept of prediction error (PE) in neuroscience. Our reasoning block uses convolution to extract abstract rules from high-level visual features of the 8 context images and generates the features of a predicted answer. PEs are then calculated between the predicted features and those of the 8 candidate answers, and are then passed to the next stage. We further integrate our novel reasoning blocks into a residual network and build a new **Pred**ictive **R**easoning **N**etwork (PredRNet). Extensive experiments show that our proposed PredRNet achieves state-of-the-art average performance on several important RPM benchmarks. PredRNet also shows good generalization abilities in a variety of out-of-distribution scenarios and other visual reasoning tasks. Most importantly, our PredRNet forms low-dimensional representations of abstract rules and minimizes hierarchical prediction errors during model training, supporting the critical role of PE minimization in visual reasoning. Our work highlights the potential of using neuroscience theories to solve abstract visual reasoning problems in artificial intelligence. The code is available at https://github.com/ZjjConan/AVR-PredRNet.



(a) From the RAVEN dataset **(b) From the PGM dataset**

Figure 1: An illustration of typical RPM questions from (a) the RAVEN (Zhang et al., 2019a) and (b) the PGM (Barrett et al., 2018) datasets. In both datasets, eight context images are provided. The goal of each RPM is to choose the correct one (highlighted in red) from eight answer images to fill in the missing one (denoted by **?**), making three rows or three columns with similar patterns. Obviously, a subject should recognize diverse visual objects, and then discover abstract relationships among these objects for inference.

1. Introduction

Deep neural networks (DNNs) (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016; Hochreiter & Schmidhuber, 1997) have shown superior performance in a variety of tasks, such as text classification (Conneau et al., 2016), machine translation (Vaswani et al., 2017; Bahdanau et al., 2015), image restoration (Kim et al., 2016; Zhang et al., 2017a), object classification and detection (Deng et al., 2009; Ren et al., 2015; Liu et al., 2016), video understanding (Karpathy et al., 2014; Tran et al., 2015; Donahue et al., 2015), and visual question answering (Antol et al., 2015; Lu et al., 2016; Fukui et al., 2016). Despite their great success in these tasks, DNNs have long been criticized for falling short of abstract reasoning, which is a hallmark of human intelligence.

[1]Sun Yat-sen University, China [2]University of Electronic Science and Technology of China, China [3]Beijing Normal University, China [4]Shanghai Jiao Tong University, China. Correspondence to: Ru-Yuan Zhang <ruyuanzhang@sjtu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

In human psychology, a widely used test of intelligence quotient is the Raven's Progressive Matrices (RPMs) (Raven & Court, 1938). Figure 1 shows two typical RPM questions from the RAVEN (Zhang et al., 2019a) and PGM (Barrett et al., 2018) datasets. An observer is asked to select the correct answer from 8 candidate answers to fill in the missing 9-th panel (denoted by **?**), where the three rows or the three columns can form the same abstract rule (*e.g.*, color or size progression). The viewer must first recognize the visual signatures of objects, such as shapes and locations, and then infer the abstract relationships between the contextual images. As such, RPMs are well suited for assessing abstract visual reasoning abilities in both humans and machines.

In cognitive science, RPMs have traditionally been addressed by symbolic models (Carpenter et al., 1990; Lovett et al., 2009; 2010; Lovett & Forbus, 2017) and more recently by DNNs (Zhang et al., 2019b; Zhuo & Kankanhalli, 2020; Hu et al., 2021; Wang et al., 2020; Jahrens & Martinetz, 2020; Hochreiter & Schmidhuber, 1997; Benny et al., 2021; Zheng et al., 2019; Spratley et al., 2020; Zhang et al., 2021; Wu et al., 2020; Mondal et al., 2022; Zhang et al., 2022). These studies focus on how to effectively discover different levels of internal statistical patterns for RPMs. To facilitate studies in this community, four famous datasets have been constructed, including RAVEN (Zhang et al., 2019a), RAVEN-FAIR (Benny et al., 2021), Impartial-RAVEN (Hu et al., 2021) and PGM (Barrett et al., 2018). Although some existing methods show superior performance on a subset of datasets, few studies show impressive performance on all datasets and different generalization cases.

Inspired by the well-known concept of prediction error (PE) in neuroscience, we develop a novel reasoning block - **P**redictive **R**easoning **B**block (PRB) - that mimics the prediction and matching process in the reasoning process of RPMs. Our PRB first predicts features from 8 context images and then encodes the differences (*i.e.,* PEs) between these predicted features and those of 8 candidate answers. The PEs are then passed to the next processing stage. We integrate our novel PRB into a residual network and construct a new network architecture called PredRNet. PredRNet shows state-of-the-art average performance on several benchmarks and superior generalization capabilities.

2. Related work

Raven Progressive Matrices. As one of the standard general IQ tests, RPM or RPM-style questions are a useful tool for understanding human abstract and analogical reasoning abilities (Raven & Court, 1938). Traditional RPMs used in psychology are designed by psychology experts and are unsuitable for modern machine learning and computer vision research. To accelerate relevant research in machine learning, Wang & Su (2015) used first-order logic to formulate RPMs, and automatically generated a large number of RPMs. Based on this dataset, Hoshen & Werman (2017) proposed the first neural network to solve simple geometric patterns in RPMs. A relation module was introduced for DNNs to learn abstract relations (Santoro et al., 2017). Besides the development of networks, the two more advanced datasets — PGM (Barrett et al., 2018) and RAVEN (Zhang et al., 2019a) were also established. Later works allow neural network models to explore row-wise and column-wise relationships (Zhang et al., 2019b; Zhuo & Kankanhalli, 2020; Zheng et al., 2019; Hu et al., 2021), discover patterns by multi-scale networks (Benny et al., 2021; Jahrens & Martinetz, 2020), improve relation modules (Spratley et al., 2020; Mondal et al., 2022), design neuro-symbolic representation (Zhang et al., 2021; 2022), and fuse features by graph networks (Wang et al., 2020). Some studies recently pointed out the defects of the original RAVEN dataset and proposed two other variants — RAVEN-FAIR (Benny et al., 2021) and Impartial-RAVEN (Hu et al., 2021).

Prediction Error in Neuroscience. Prediction error is a well-known concept in neuroscience. Schultz et al. (1997) pioneered the study of reward prediction error and showed that the difference between the predicted reward and the actual reward received is the key factor driving biological learning. This neural substrate fits well with the temporal difference learning proposed in the field of reinforcement learning. A similar concept was later introduced into sensory processing. Rao & Ballard (1999) incorporated prediction error into a three-layer neural network and found that the neural receptive fields after training showed strong similarities to the center-surround effects reported in the neurophysiological literature (Srinivasan et al., 1982; Dan et al., 1996; Hubel & Wiesel, 1968; Bolz & Gilbert, 1986; Desimone & Schein, 1987). The concept of prediction error has now been extended to the auditory system (Smith & Lewicki, 2006), the hippocampus (Mehta, 2001), and the prefrontal cortex (Summerfield et al., 2006).

Friston & Kiebel (2009) further proposes a unifying theoretical framework for understanding human cognition. Namely, the brain constructs an internal model to approximate the operations of the external environment. This internal model generates predictions about what the observed sensory evidence should be, and the brain uses prediction errors to update the belief held in the internal model. This theory can explain a wide range of cognitive phenomena, including binocular rivalry (Hohwy et al., 2008), reinforcement learning (Alexander & Brown, 2018), visual illusions (Pang et al., 2021), and even atypical behavior in psychiatric populations (Sterzer et al., 2019) *etc*. In summary, prediction error is one of the most fundamental neuroscientific concepts and may make a significant contribution to others.

Prediction Error in Computer Vision. Prediction-based

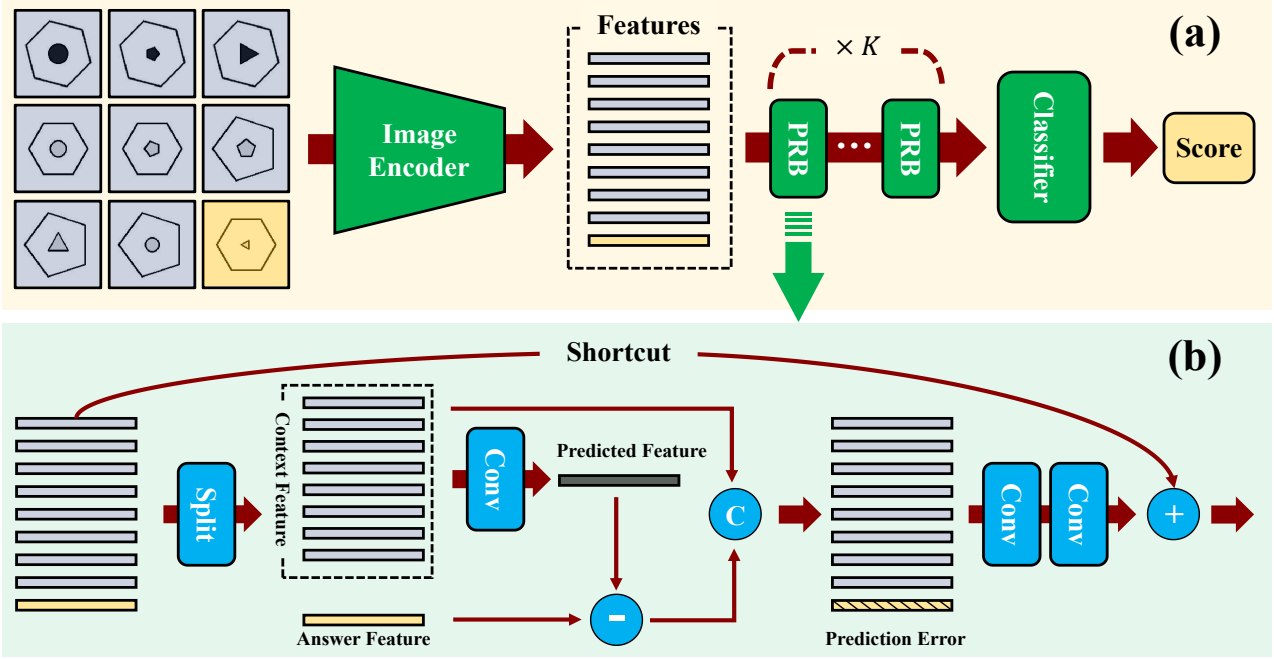


Figure 2: An overview of our PredRNet. (a) PredRNet contains an image encoder to extract input image features, multiple stacked Predictive Reasoning Blocks (PRBs) to abstract relationships between features from context images and answer images, and a classifier to output eight scores for each RPM question. (b) Details of the proposed reasoning block – PRB.

processing has also been introduced into data compression (Schmidhuber & Heil, 1996; Atal & Schroeder, 1970; Schmidhuber & Heil, 1995). In the computer vision community, some works have also used prediction-based processing as loss functions or training strategies for audio or object recognition (Henaff et al., 2011; Doersch et al., 2015; Kavukcuoglu et al., 2010; Gregor & LeCun, 2010; Choksi et al., 2021; Wen et al., 2018; Zhang et al., 2017b; Oord et al., 2018). For example, Gregor & LeCun (2010) trained a predictor to approximate the original sparse codes to improve inference speed. Doersch et al. (2015) proposed an unsupervised learning method by predicting the relative position of image patches. Zhang et al. (2017b) designed a framework for learning good representations by estimating color images from grayscale images. Instead of using PE only as a loss function or a training strategy, Choksi et al. (2021); Wen et al. (2018); Lotter et al. (2016) incorporate PE into the network architecture for object recognition. Although our work uses PE like several previous studies, we focus on abstract visual reasoning tasks and do a very different implementation. First, our network performs cross-image prediction, whereas previous methods only perform prediction within a single image. The two structures are different, but all satisfy the prediction-based framework. Second, due to the nature of the problem, our model only fuses high-level features across images, whereas Choksi et al. (2021); Wen et al. (2018); Lotter et al. (2016) emphasize the computation of prediction errors across all layers. Furthermore, our model iterates prediction-based processing in a stacked fashion without any recurrent connection.

3. Methods

The structure of our PredRNet is shown in Figure 2. It consists of three components: (1) an **Image Encoder** to transform each image into a 3-dimensional high-level representation (features), (2) multiple ($K \geq 2$) stacked **Predictive Reasoning Block (PRB)** to extract relationships between the representations of context and answer images, and (3) a **Classifier** to output the scores for 8 answer images. In each RPM, the answer image with the highest score is selected as the final answer.

Image Encoder. A ResNet architecture is used as our image encoder (He et al., 2016). Several previous studies have provided some baseline results based on the popular ResNet-18 or ResNet-50 networks and their extended variants (hereafter referred to as baseline networks) (Zhang et al., 2019a;b; Barrett et al., 2018; Hu et al., 2021). For example, SRAN (Hu et al., 2021) combines three ResNet-18 to extract features and then uses their adapted structure to discover rules. We argue that these networks are suboptimal because their properties (*e.g., large kernel sizes, more stacked blocks,* $32\times$ *subsampling*) are designed for natural images. In RPM questions, the objects are relatively small (see Figure 1) and more difficult to detect (Lin et al., 2014). In addition, some

of these baseline networks combined all the images of an RPM question at the first layer. This "early fusion" explores only the low-level relationships between images rather than the high-level relationships, which reduces the reasoning performance as our experiment shown. Therefore, we build a new ResNet to provide a strong baseline for solving RPMs.

Our image encoder has four **ResBlocks**, each containing a residual branch and a shortcut branch. The residual branch has three convolutional layers with kernel sizes of 3×3. The first convolutional layer downsamples the input features with a stride of 2 to expand the receptive fields of the neurons to extract higher-level information. Thus, the total subsampling stride of our image encoder is 16. The shortcut branch first applies an average pooling layer to the downsampled input features (Zhang, 2019), and then uses a 1×1 convolutional layer to match the output size of the residual branch. These two branches are then added together to form the next block. In total, a ResBlock can be formulated as:

$$\boldsymbol{X}^l = ReLU(BN(Conv_{3\times3}(\boldsymbol{X}^{l-1}))), l \in \{1,2,3\} \quad (1)$$

$$\boldsymbol{X}^{'} = \boldsymbol{X}^3 + BN(Conv_{1\times1}(AvgPool(\boldsymbol{X}^0))), \quad (2)$$

where $\boldsymbol{X}^0$ and $\boldsymbol{X}^{'}$ are the block input and output, respectively. $Conv_{3\times3}$ and $Conv_{1\times1}$ are convolutional layers with kernel sizes of 3×3 and 1×1, respectively. Both the first convolutional layer ($l = 1$) and the average pooling layer ($AvgPool$) downsample inputs with a stride of 2.

After the above defined 4-block structure, we append a single $BN(Conv_{1\times1}(\cdot))$ to reduce the feature dimension for further processing. For simplicity, we will henceforth refer to our image encoder as **ResNet-4B**. To extract features in an RPM problem (Figure 2a), a single answer image $\boldsymbol{I}_i^a$ is combined with the eight context images $\boldsymbol{I}_{\{1,...,8\}}^c$ to form an input, denoted as $\boldsymbol{I}_i = [\boldsymbol{I}_1^c, \boldsymbol{I}_2^c, ..., \boldsymbol{I}_8^c, \boldsymbol{I}_i^a], \boldsymbol{I}_i \in \mathcal{R}^{9\times1\times80\times80}, i \in \{1, ..., 8\}$. Where i is the index of the answer image. $\boldsymbol{I}_i$ is then passed sequentially through ResNet-4B to obtain nine sets of image features, denoted as $\boldsymbol{X}_i = [\boldsymbol{X}_1^c, \boldsymbol{X}_2^c, ..., \boldsymbol{X}_8^c, \boldsymbol{X}_i^a], \boldsymbol{X}_i \in \mathcal{R}^{9\times32\times5\times5}$. Importantly, this feature extraction step is performed in parallel for all images without considering any rule-based relationships between them. After feature extraction, $\boldsymbol{X}_i$ is the i-th answer-related features, including all features from the eight context images and the i-th answer image.

Stacked Predictive Reasoning Blocks (PRBs). Although solving RPMs is a high-level cognitive task, we argue that it still follows the prediction-and-matching process (Spratling, 2016). In an RPM problem, an observer must first examine all 8 context images to learn the implicitly embedded rule. According to the learned rule, she then makes a prediction of what the correct answer should be, and matches this prediction to the 8 answer images. The matching step can be formulated as calculating the error between the prediction and the answer images. And the prediction errors are in turn used to refine the learned rule. The initial learned rule and prediction may be incorrect, so the prediction and matching process should be iterated. In this scenario, the prediction error is the critical cue for correct reasoning.

Based on the above theory, we implement PRB as shown in Figure 2 (b). The feature map $\boldsymbol{X}_i$ is first transformed from $[9, 32, 5, 5]$ to $[9, 32, 1, 25]$, followed by a transpose operator to form a tensor of size $[1, 32, 9, 25]$. The above tensor reformulation allows us to use simple 2D convolutional layers to extract useful spatial cues along the 4-th dimension and rule relationships along the 3-rd dimension, greatly reducing the implementation effort. Specifically, we split the current $\boldsymbol{X}_i$ into two feature sets along the 3-rd dimension: the eight context features $\boldsymbol{X}^c \in \mathcal{R}^{1\times32\times8\times25}$ and the i-th answer feature $\boldsymbol{X}_i^a \in \mathcal{R}^{1\times32\times1\times25}$. Note that these features only belong to their corresponding images, because all images are processed in parallel by our ResNet-4B. After splitting, an 8×1 convolutional layer with 32 channels is used to aggregate all the context features $\boldsymbol{X}^c$ into a single predicted feature map $\boldsymbol{X}_i^p \in \mathcal{R}^{1\times32\times1\times25}$, which has the same size as the answer features $\boldsymbol{X}_i^a$. The prediction error is then obtained by subtracting the predicted features from the original answer features. We show the generation of the prediction error as follows:

$$\boldsymbol{X}_i^p = BN(Conv_{8\times1}(\boldsymbol{X}^c)) \quad (3)$$

$$\boldsymbol{E}_i^a = ReLU(\boldsymbol{X}_i^a) - ReLU(\boldsymbol{X}_i^p) \quad (4)$$

Currently, the prediction error $\boldsymbol{E}_i^a \in \mathcal{R}^{1\times32\times1\times25}$ stores the discrepancy between the prediction from the context features and the i-th answer features. We then concatenate such a predicted error $\boldsymbol{E}_i^a$ with the original context features $\boldsymbol{X}^c$ along the 3-th dimension, and pass these combined features with another two convolutional layers. In addition, similar to our ResBlock (Eq. (1) and Eq. (2)), a shortcut branch is added to facilitate the optimization. Therefore, we can formulate this process as follows:

$$\boldsymbol{Y}_i^0 = [\boldsymbol{X}^c; \boldsymbol{E}_i^a], \quad \boldsymbol{Y}_i^0 \in \mathcal{R}^{1\times32\times9\times25} \quad (5)$$

$$\boldsymbol{Y}_i^l = ReLU(BN(Conv_{3\times3}(\boldsymbol{Y}_i^{l-1}))), \quad l \in \{1,2\} \quad (6)$$

$$\boldsymbol{Y}_i^{'} = \boldsymbol{Y}_i^2 + BN(Conv_{1\times1}(\boldsymbol{X}_i)), \quad (7)$$

where $[\cdot;\cdot]$ is the concatenation operation. $\boldsymbol{Y}_i^{'}$ is the current PRB output with the same size of features $\boldsymbol{X}_i$. By far, our PRB extracts the abstract relations for an RPM question. However, a single prediction and matching process is unlikely to capture the abstract rules very accurately. Similar to the human mental process, such a prediction-and-matching process can be iterative, and the learned rule can be gradually refined and tested. Therefore, a stacked structure is

designed by combining several ($K \geq 2$) PRBs, as shown in the middle part of Figure 2 (a). As mentioned before, our PRB forwards both errors and context features for further processing. This allows the last two convolutional layers to refine contextual features and prediction errors by aggregating cues along the last two dimensions. As a result, our block allows context features to be adapted to current errors and then become more effective in the next stage to generate prediction errors. By incrementally propagating both context features and prediction errors, stacked PRBs are able to incrementally extract abstract rules.

Classifier. Similar to previous models (Benny et al., 2021; Zhang et al., 2019b;a; Spratley et al., 2020), our classifier contains two fully-connected layers to output a single score for the i-th answer. A batch normalization layer and a ReLU function are added between these fully-connected layers. In sum, for each RPM question, our PredRNet processes eight answer images in parallel and outputs eight scores.

4. Experiments

We first compare our PredRNet with many state-of-the-art methods on four popular abstract visual reasoning datasets, including PGM (Neutral)(Barrett et al., 2018), RAVEN (Zhang et al., 2019a), I-RAVEN (Hu et al., 2021), and RAVEN-FAIR (Benny et al., 2021). Most previous studies did not evaluate their methods on all of these datasets. For a fair and comprehensive comparison, we run their published codes on all four datasets. In addition, to demonstrate the generalizability of our PredRNet, we also perform experiments on all OOD versions of PGM, CLEVR-Matrices (Mondal et al., 2022) and two other human-like reasoning tasks, Bongard-LOGO (Nie et al., 2020) and Visual Analogy Data (VAD) (Hill et al., 2019). In the following, we first detailed describe all datasets and our implementations. Then we compare the proposed PredRNet with other state-of-the-art methods. Finally, we present in-depth studies.

4.1. Datasets & Implementations

PGM (Barrett et al., 2018) contains 8 different sub-datasets, each having 1,222,000 questions with 119,552,000 images. It has diverse abstract rules (*i.e.,* XOR, OR, Progression and AND) among objects. We mainly compare PredRNet with other state-of-the-arts on the *Neutral* sub-dataset. Other sub-datasets are included to examine the Out-Of-Distribution generalization of our PredRNet.

RAVEN (Zhang et al., 2019a) introduces a different set of relationships, including progression, constant, union, and arithmetic calculations. This dataset includes 7 distinct configurations, *i.e.,* Center, 2x2Grid, 3x3Grid, Left-Right, Up-Down, Out-InCenter, and Out-InGrid. Each configuration contains 10,000 questions, yielding a total of 70,000 questions with 1,112,000 images. However, a few studies (Hu et al., 2021; Benny et al., 2021; Spratley et al., 2020) point out that this original version of RAVEN has been recently shown problematic. RAVE-FAIR and I-RAVEN (see below) are proposed to correct the bias.

RAVEN-FAIR (Benny et al., 2021) and **Impartial-RAVEN** (Hu et al., 2021) are recently developed to fix the bias in RAVEN. Both of them contain the same context images as RAVEN but differ in the way to generate negative answers. In RAVEN-FAIR, negative answers are iteratively generated by randomly changing one attribute of the true answer. While Impartial-RAVEN uses a bisection tree to modify one attribute at a time, but in different attribute direction. (Benny et al., 2021) and (Hu et al., 2021) show that RAVEN-FAIR and I-RAVEN are better in evaluating models.

CLEVR-Matrices (Mondal et al., 2022) is another RPM-like dataset based on the widely used visual question answer dataset – CLEVR (Johnson et al., 2017). This dataset includes 3 distinct configurations, *i.e.,* Logic, Location, and Count. Each configuration has 20,000 questions, including 16,000 for training, 2,000 for validation, and 2,000 for testing. For questions, this dataset has three kinds of visual attributes, *i.e.,* shape, size, and color, and rules are independently sampled from the set of {null, constant, distribution-of-3}. To reduce the biases in the original RAVEN dataset (Zhang et al., 2019a), answer choices are generated using the attribute bisection tree algorithm proposed in (Hu et al., 2021). For the evaluation, model is trained jointly on all three configurations as suggested in (Mondal et al., 2022).

Bongard-LOGO (Nie et al., 2020) and **Visual Analogy Data (VAD)** (Hill et al., 2019) are another datasets for evaluating human-level visual reasoning. For Bongard-LOGO, it mimics the Bongard problem that reasons visual concepts from their contexts with a few examples. Therefore, Bongard-LOGO transforms 12,000 problems into a few-shot binary classification task. For VAD, it mainly contains five subdatasets, each containing around 600,000 problems. This dataset is employed to check whether our PredRNet can learn to make analogy in visual domain.

Implementations. For our ResBlocks, we set the filters to [32,64,96,128] from the first to the last block. For our PRB, the $Conv$ in Eq.(3) with kernel sizes of [1,32,8,1] generates a prediction with the same shape of answer features, *i.e.,* $\boldsymbol{X}_i^a \in \mathcal{R}^{1\times 32\times 1\times 25}$. The filters of the other two $Conv$s are set to 128 and 32, respectively. In addition, we add three (K=3) PRBs after our image encoder because this gives the best overall performance on the validation sets. The effect of K will be discussed in later sections.

All datasets have training, validation, and test sets. The validation set is used to select the best checkpoint for evaluation. Our model accepts 80×80 images as input. Optimization

Table 1: Recognition accuracy (%) on PGM Neutral (PGM-N), original RAVEN (RVN-O), RAVEN-FAIR (RVN-F), and Impartial-RAVEN (I-RVN). For all RAVENs, accuracy is obtained by averaging across all seven configurations. † indicates the performance was not reported in their original paper, and is obtained by running their published codes. The best and the second best results on each dataset are highlighted by **bold** and underline, respectively. Our PredRNet obtains the state-of-the-art average (Avg) performance on all the four compared datasets.

Method	WReN	LEN	CoPINet	SRAN	DCNet	MLRN	SCL	MXNet	Rel-Base	MRNet	STSN	PredRNet
PGM-N	62.6	68.1	56.4	71.3	68.6	98.0	88.9	66.7	85.5	94.5	**98.2**	97.4
RVN-O	16.8	72.9	91.4	54.3†	93.6	12.3†	91.6	83.9	91.7	**96.6**	89.7†	95.8
RVN-F	30.3	51.0	50.6	72.9†	56.1†	29.5†	90.1†	35.1†	93.5†	88.4	95.4†	**97.1**
I-RVN	23.8	41.4	46.1	60.8	47.2†	12.3†	95.0	26.8†	91.1†	83.5†	95.7	**96.5**
Avg	33.4	58.4	61.1	64.8	66.4	38.0	91.4	53.1	90.5	90.8	94.8	**96.7**

is done by the Adam solver (Kingma & Ba, 2015) with a learning rate of 1e-3 and a batch size of 128. The weight decay is 1e-5 for most of the tested datasets, and 1e-7 for the PGM datasets because PGM is significantly larger than other datasets. It is worth noticing that we do not include other supervision signals (*e.g.*, metadata) during training. In addition, for each RAVEN, we report the median result from 3 different runs. For all 8 sub-datasets of PGM, since each of them has about 1.2×16 million images, training on such large-scale datasets will take too much computational time. Therefore, we only report a single result, similar to many previous works (Johnson et al., 2017; Zhang et al., 2019b; Benny et al., 2021). For other datasets, we follow the settings in their original papers for a fair comparison.

4.2. Main Results

State-of-the-art Comparisons. Here, we compare the proposed PredRNet with several previous models, including WReN (Barrett et al., 2018), LEN (Zheng et al., 2019), CoPINet (Zhang et al., 2019b), SRAN (Hu et al., 2021), DCNet (Zhuo & Kankanhalli, 2020), SCL (Wu et al., 2020), MLRN (Jahrens & Martinetz, 2020), MXGNet (Wang et al., 2020), Rel-Base (Spratley et al., 2020), MRNet (Benny et al., 2021) and STSN (Mondal et al., 2022). Some baseline methods such as LSTM (Zhang et al., 2019a), ResNet (Zhang et al., 2019a), CNN (Zhang et al., 2019a) and ResNet-DRT (Zhang et al., 2019a) are not included because they perform significantly worse than these methods. Experiments are performed on three RAVENs and PGM *Neutral*.

Table 1 shows all the results. We reach three main conclusions. First, our PredRNet achieves the best average performance on the four datasets. Specifically, STSN introduces slot attention (Locatello et al., 2020) to extract image-wise features and then proposes a transformer-based module to explore relationships between contexts and choices for reasoning. Because of the powerful attention operations, STSN provides the best average performance (94.8%) among all compared method. Our proposed method, PredRNet, outperforms STSN with an average performance of 96.7%. In addition, our PredRNet actually achieves better performance on the Impartial-RAVEN (+0.8%) and RAVEN-FAIR (+1.7%) datasets than this second-place performer, respectively. Our model is also very competitive against MLRN and STSN on PGM, and MRNet on RAVEN. Second, some recently proposed methods, such as MLRN, DCNet, and CoPINet, only show good results on one or two benchmark datasets. For example, MLRN nearly obtains perfect result (98%) on PGM, but performs poorly on the three RAVENs (all <30). Both DCNet and CoPINet achieve very promising results on the original RAVEN, but unsatisfactory results on the other three benchmarks. In contrast, our PredRNet achieves good performance on all four benchmarks (all > 95). These results clearly demonstrate the robustness of our PredRNet in discovering different types of rules in different datasets. Third, beside the STSN, SCL, Rel-Base, and MRNet are three competitive models, although they do not perform as well as ours. SCL and Rel-Base directly extract relations in all eight context images, without special designs for row and column rules. MRNet, on the other hand, deliberately includes row and column relation modules. Similar to SCL and Rel-Base, our PredRNet does not explicitly distinguish between row-wise and column-wise relations. Instead, the key component of our PredRNet is to use prediction errors as our processing signals. This special design helps our method to improve the performance up to 96.6% on average. All these results show that our PRB serves as an efficient module to discover high-level abstract relations.

Out-Of-Distribution Generalization in PGM. We further evaluate the out-of-distribution (OOD) generalization capability of PredRNet. We run our PredRNet directly on all subdatasets without changing the network architecture. The results are shown in Table 2a. Our PredRNet achieves the best average OOD generalization results across all subdatasets. In particular, our model achieves the best results compared to others in the two most commonly used subdatasets – interpolation and extrapolation. Table 2a suggests that PredRNet has good OOD generalization capabilities.

CLEVR-based RPM. We also conduct an experiment on

Table 2: Performance on other kinds of evaluation.

(a) Recognition accuracy (%) on all regimes of PGM (1 *Neutral* and 7 OOD subsets, Ntr: Neutral, Int: Interpolation, Ext: Extrapolation, H.O: Held-Out, P: Pairs, TP: TriplePairs, LT: LineType, SC: ShapeColor). The best and the second best results are highlighted using **bold** and underline. Our PredRNet obtains competitive results without using any extra supervision signals.

Method	Ntr	Int	Ext	H.O.P	H.O.TP	H.O.T	H.O.LT	H.O.SC	Avg
WReN	62.6	64.4	17.2	27.2	41.9	19.0	14.4	12.5	32.4
MXGNet	66.7	65.4	18.9	33.6	43.3	19.9	16.7	16.6	35.1
MRNet	93.4	68.1	19.2	38.4	55.3	**25.9**	**30.1**	**16.9**	43.4
PredRNet	**97.4**	**70.5**	**19.7**	**63.4**	**67.8**	23.4	27.3	13.1	**47.1**

(b) Recognition accuracy (%) on all configurations of CLEVR-Matrices (Mondal et al., 2022). The best and the second best results are highlighted using **bold** and underline. Our PredRNet obtains competitive results.

Method	Logic	Location	Count	Avg
MLRN	47.4	21.4	23.6	30.8
SCL	80.9	65.8	64.9	70.5
STSN	99.2	**100.0**	99.6	99.6
PredRNet	**100.0**	99.5	**99.9**	**99.8**

(c) Performance on the few-shot problem of the Bongard-LOGO dataset (Nie et al., 2020). Testing accuracy (%) on different splits are reported (*i.e.,* free-form shape (FF), basic shape (BA), combinatorial abstract shape (CM), and novel abstract shape (NV)). Base-SC, Base-MoCo and ProtoNet are three best performers according to (Nie et al., 2020). Our PredRNet achieves the state-of-the-art results across all splits.

Method	FF	BA	CM	NV
Base-SC	66.3±0.6	73.3±1.3	63.5±0.3	63.9±0.8
Base-MoCo	65.9±1.4	72.2±0.8	63.9±0.8	64.7±0.3
ProtoNet	64.6±0.9	72.4±0.8	62.4±1.3	65.4±1.2
PredRNet	**74.6±0.3**	**75.2±0.6**	**71.1±1.5**	**68.4±0.7**

(d) Performance on Visual Analogy Data (Hill et al., 2019). Shekhar & Taylor (2021) is the leading method proposed in recent year. The "learning-by-contrast (lbc)" is used for all subsets. Our PredRNet achieves the state-of-the-art average results.

Method	LBC (2019)	NSM (2021)	PredRNet
Extrapolation	0.62±0.020	**0.74**	0.72±0.060
Interpolation	0.93±0.004	0.93	**0.97±0.002**
N.D.Transfer	0.87±0.005	0.88	**0.96±0.003**
N.D.ShapeColor	0.78±0.004	0.78	**0.80±0.010**
N.D.LineType	0.76±0.020	0.79	**0.82±0.010**
Avg	0.79	0.82	**0.85**

the recently introduced CLEVR-Matrices (Mondal et al., 2022) dataset, which contains more object attributes and rules. This dataset is similar to all RAVEN datasets, but more focuses on the rendered 3D shapes in a scene. We thus follow the original paper (Mondal et al., 2022) and directly train our proposed PredRNet jointly on three configurations. Training settings are similar to the ones used in our RAVEN experiments. The results are shown in Table 2b. Our PredRNet performs very competitively with the attention-based method – STSN, which uses slot attention and self-attention for feature extraction and relationship modeling, respectively.

Human-like Reasoning Tasks. To test the generality of our model beyond RAVEN tasks, we evaluate PredRNet on two additional benchmarks - Bongard-LOGO (Nie et al., 2020) and Visual Analogy Data (VAD) (Hill et al., 2019). These two datasets are thought to reflect more human-like reasoning processes. On Bongard-LOGO, we follow that paper to solve a 2-way 6-shot few-shot classification problem. In PredRNet, we compute the error of each query image from 6 support images. We then use default settings (without using symbolic information) to train (Nie et al., 2020), and report the results in Table 2c. Clearly, PredRNet achieves leading performance in all subsets.

VAD problems are similar to RAVEN problems. A VAD problem contains 5 context images and 4 candidate images. We combine each candidate image and 5 context images to form an answer-related group, and feed 4 groups in each problem into our PredRNet. The number of training epochs is set to 3 for all subdatasets as suggested in (Webb et al., 2020). The results in Table 2d show that our proposed PredRNet obtains the best results among all compared methods.

In summary, all the above results show that our proposed PredRNet is effective and flexible to solve various forms of abstract visual reasoning problems.

4.3. Ablation Experiments

Different Image Encoders. As in our previous presentation, some of the existing methods used the popular ResNet-18 and ResNet-50 as baseline methods for comparison. These baseline methods fuse all images from the first convolutional layer (*i.e.*, early fusion). We argue that the images should instead be processed in parallel and transformed into high-level feature embeddings, and then a reasoning algorithm should take place and process their relationships (*e.g.*, late fusion). Thus, we provide stronger baseline results by using late fusion for these baseline methods. We also include our image encoder – ResNet-4B. Although the image encoder is not our contribution here, we would like to provide some empirical results which might be helpful to this community.

All comparison results are shown in Table 3. The early

Table 3: Recognition accuracy (%) of different image encoders and blocks on all three RAVEN datasets. ResNet-XE (RN-XE) and ResNet-XL (RN-XL) indicate different fusion methods, with E for early fusion at the first convolutional layer and L for late fusion in the output of image encoders.

Method	RVN-O	RVN-F	I-RVN	Avg
Different Image Encoders				
ResNet-18E	58.0	15.8	15.8	29.9
ResNet-50E	61.8	17.3	11.8	30.3
ResNet-18L	53.7	77.6	54.2	61.8
ResNet-50L	68.1	62.7	68.7	66.5
ResNet-4B	57.3	75.6	71.5	68.1
ResNet-4B + Additional K ResBlocks				
$K = 1$	59.0	76.5	62.0	65.8
$K = 2$	60.1	77.6	56.6	64.8
$K = 3$	59.5	74.3	49.2	60.9
$K = 4$	56.6	72.2	41.7	56.8
ResNet-4B + Additional K PRBs				
$K = 1$	94.6	95.8	95.0	95.1
$K = 2$	95.5	96.4	96.5	96.1
$K = 3$	95.8	97.1	96.5	96.5
$K = 4$	96.0	96.8	94.8	95.8

fusion encoders (denoted by E) perform significantly worse than the late fusion encoders (denoted by L). For example, ResNet-18E and ResNet-50E perform only slightly better than chance (12.5%) on RAVEN-FAIR and I-RAVEN. In contrast, ResNet-18L and ResNet-50L achieve much better performance. In addition, the larger number of model parameters in ResNet-50L does not lead to a significant overall performance improvement. Our ResNet-4B contains only 1.28 M parameters (*v.s.* 23.8 M in ResNet-50L), but achieves the best overall performance. In addition, according to Table 1 and Table 3, ResNet-4B, even as a baseline, outperforms many existing models by a wide margin.

ResBlocks *v.s.* PRBs. In this section, we use our ResNet-4B as the image encoder and append different numbers ($K \in [1, 4]$) of PRBs to evaluate the effect of PRBs. It is also interesting to compare PRBs with the case where our ResBlock Eq. (1) is appended to Eq. (2).

All comparison results are shown in Table 3. Interestingly, we find that simply adding more ResBlocks degrades the overall performance. For example, adding four additional ResBlocks results in the worst performance on all datasets compared to ResNet-4B. We speculate that in the ResNet-4B baseline, the image encoder processes the 16 images in parallel, and only the classifier combines the features across images to extract their relationships. Thus, simply adding ResBlocks may help to process the features in individual images rather than extracting cross-image rules. Instead, our PRB explores the cross-image information by the method similar to the calculation of PE. Therefore, PRBs can greatly improve the reasoning performance over the baseline ResNet-4B. For example, adding a single PRB to the ResNet-4B baseline improves performance by about 20% for RAVEN-FAIR and about 24% for Impartial-RAVEN. Adding 2 or 3 PRBs further improves performance on all datasets, giving new state-of-the-art results. We find that adding $K = 4$ PRBs leads to a slight drop in performance. This phenomenon may be due to overfitting.

Table 4: Ablation experiments on different operations in our PRB. MLPs: replacing PRBs with MLPs. F-MLPs: replacing Eqn.(4) with MLPs. Fwd.Err: only forwarding errors. Rmv.PC: removing prediction errors. $\delta(x) + \delta(y)$: replacing Eqn.(4) with $ReLU(x) + ReLU(y)$.

Method	RVN-O	RVN-F	I-RVN
ResNet-4B	57.3	75.6	71.5
MLPs	58.1	70.4	66.1
F-MLPs	68.9	91.3	64.9
Fwd.Err	90.2	94.2	93.1
Rmv.PC	92.3	93.6	92.1
$\delta(x) + \delta(y)$	94.9	96.7	95.0
Ours	**95.8**	**97.1**	**96.5**

Different Operators in PRBs. We also perform several important ablation experiments and show all the results in Table 4. First, we remove the step of computing prediction errors, so that our PRB contains only the two convolutional layers for extracting relationships (denoted as Rmv.PC). We find that removing the prediction errors impairs the performance of the model, indicating the central role of prediction errors. Second, our implementation here can also be explained by attention, which predicts that addition and subtraction make no difference in processing. We therefore replace Eq.(4) with $ReLU(x) + ReLU(y)$ like an attention operator. In this case, (+) and (-) have completely different mathematical effects. We find that subtraction performs better, further supporting the effects of prediction error rather than prediction summation or attention. Third, we also attempt to pass errors directly without further refinement and find a significant drop in performance. These results show that our design of refining both context features and errors produces more error-aware context embedding features, which in turn produce more appropriate errors for the next stage. Finally, we directly replace our PRB with MLP to extract relationships between context and response features, and obtain the worse performance. Overall, all these results confirm the positive contribution of the operations in PRB.

4.4. Representations of Rule-related Feature Attributes

Although PredRNet achieves impressive performance on a variety of benchmarks, it remains unclear whether pre-

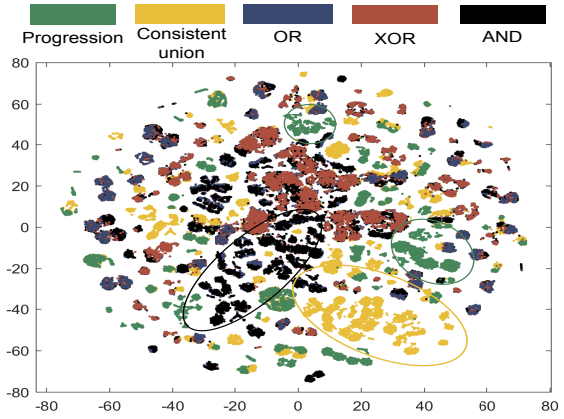


Figure 3: T-SNE of abstract rules (*e.g.,* progression, AND, XOR) in PGM in the 3-rd PRB. The clustered embeddings of the same abstract rule indicate that PredRNet is powerful at discovering rules. Solid circles illustrate different clusters.

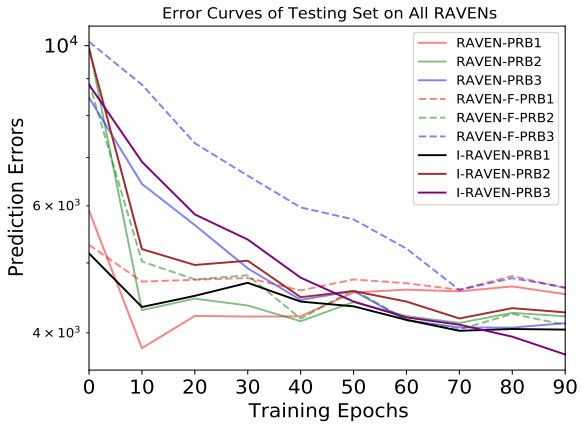


Figure 4: Prediction errors calculated by Eqn. (4) on the test sets of all RAVENs along the training progression. PRB-K indicates the K-th block. Overall, our proposed PredRNet indeed minimizes these errors during the training process.

diction errors encode abstract rules. In an RPM question, an observer must identify the rule regardless of detailed visual attributes. We therefore analyze the low-dimensional embedding of abstract rules in prediction errors in the final PRB using t-SNE (Van der Maaten & Hinton, 2008). Specifically, we extract all errors from the 3-rd PRB block on the test sets of the PGM *Neutral* dataset. Because each RPM contains $1 \sim 4$ rules, it is not easy to visualize a data point with different rules. For simplicity, we group all RPM questions according to their relation types, *i.e.,* progression, AND, OR, XOR and consistent_union. We then use these relation types as labels to show all errors. As shown in Figure 3. We find clustered representations of abstract rules, indicating that our PredRNet can robustly identify important visual attributes and rules. The relative overlap between XOR and AND indicates the difficulty of dissecting the two rules. Note that these representations can be further refined by subsequent MLPs to obtain a correct answer.

4.5. PredRNet Minimizes Prediction Errors

Finally, we seek to understand whether PredRNet minimizes prediction errors. To confirm this notion, we compute the prediction errors in the three PRBs on the test set during model training (see Figure 4). We find that prediction errors decrease as training proceeds, suggesting that our network is indeed learned to minimize prediction errors during visual processing. It is noteworthy that in PredRNet, prediction errors are not included as a loss term. In other words, we do not explicitly train the model to minimize prediction errors.

5. Limitation

Our PredRNet has two limitations. First, the error operator in our model slightly improves model performance as compared to the attention-based mechanism. This suggests that our current PredRNet may not fully explore the power of error computation. Future studies could be done by learning more rule-related cues with a good regularization (*e.g.,* MSE loss). Second, the human reasoning process and our model are not strictly identical. Our model is fully supervised, training on the full set of benchmarks. However, humans can answer RPM questions after learning only a few examples. We acknowledge such differences and argue that abstract reasoning in DNNs still lags behind humans. We have only made progress in the supervised learning setting. Future studies could consider building semi-supervised, unsupervised, or few-shot learning models of reasoning.

6. Summary

In this work, we exploited the idea of prediction error in neuroscience and proposed a novel predictive reasoning neural network to extract abstract relations. Our reasoning block first used convolution to extract abstract rules' features from 8 context images and then generated predictions. We then calculated errors between these predictions and those of the 8 candidate answers. We further integrated this block into a residual network to for our PredRNet. Experimental results show improved robustness and generalization capabilities of our PredRNet in a variety of visual reasoning scenarios.

Acknowledgements

This work was partially supported by the NSFC Projects (62206316, 32100901, 32171094), the Key-Area Research and Development Program of Guangzhou (202007030004), the Guangdong NSF Project (2022A1515011254), the Shanghai NSF Project (21ZR1434700), the Sichuan NSF Project (2022NSFSC0527), the Shanghai Pujiang Program (21PJ1407800), the Research Project of Shanghai Science and Technology Commission (20dz2260300) and the Fundamental Research Funds for the Central Universities.

References

Alexander, W. H. and Brown, J. W. Frontal cortex function as derived from hierarchical predictive coding. *Scientific reports*, 8(1):1–11, 2018.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. Vqa: Visual question answering. In *ICCV*, pp. 2425–2433, 2015.

Atal, B. S. and Schroeder, M. R. Adaptive predictive coding of speech signals. *Bell System Technical Journal*, 49(8): 1973–1986, 1970.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.

Barrett, D., Hill, F., Santoro, A., Morcos, A., and Lillicrap, T. Measuring abstract reasoning in neural networks. In *ICML*, pp. 511–520. PMLR, 2018.

Benny, Y., Pekar, N., and Wolf, L. Scale-localized abstract reasoning. In *CVPR*, pp. 12557–12565, 2021.

Bolz, J. and Gilbert, C. D. Generation of end-inhibition in the visual cortex via interlaminar connections. *Nature*, 320(6060):362–365, 1986.

Carpenter, P. A., Just, M. A., and Shell, P. What one intelligence test measures: a theoretical account of the processing in the raven progressive matrices test. *Psychological review*, 97(3):404, 1990.

Choksi, B., Mozafari, M., Biggs O'May, C., Ador, B., Alamia, A., and VanRullen, R. Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics. *NeurIPS*, 34, 2021.

Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

Dan, Y., Atick, J. J., and Reid, R. C. Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *Journal of neuroscience*, 16(10):3351–3362, 1996.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. Ieee, 2009.

Desimone, R. and Schein, S. J. Visual properties of neurons in area v4 of the macaque: sensitivity to stimulus form. *Journal of neurophysiology*, 57(3):835–868, 1987.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *ICCV*, pp. 1422–1430, 2015.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pp. 2625–2634, 2015.

Friston, K. and Kiebel, S. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences*, 364(1521):1211–1221, 2009.

Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *ICML*, pp. 399–406, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Henaff, M., Jarrett, K., Kavukcuoglu, K., and LeCun, Y. Unsupervised learning of sparse features for scalable audio classification. In *ISMIR*, volume 11, pp. 2011. Citeseer, 2011.

Hill, F., Santoro, A., Barrett, D. G., Morcos, A. S., and Lillicrap, T. Learning to make analogies by contrasting abstract relational structure. *arXiv preprint arXiv:1902.00120*, 2019.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Hohwy, J., Roepstorff, A., and Friston, K. Predictive coding explains binocular rivalry: An epistemological review. *Cognition*, 108(3):687–701, 2008.

Hoshen, D. and Werman, M. Iq of neural networks. *arXiv preprint arXiv:1710.01692*, 2017.

Hu, S., Ma, Y., Liu, X., Wei, Y., and Bai, S. Stratified rule-aware network for abstract visual reasoning. *AAAI*, 2021.

Hubel, D. H. and Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

Jahrens, M. and Martinetz, T. Solving raven's progressive matrices with multi-layer relation networks. In *IJCNN*, pp. 1–6. IEEE, 2020.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pp. 2901–2910, 2017.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. Large-scale video classification with convolutional neural networks. In *CVPR*, pp. 1725–1732, 2014.

Kavukcuoglu, K., Ranzato, M., and LeCun, Y. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv preprint arXiv:1010.3467*, 2010.

Kim, J., Lee, J. K., and Lee, K. M. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pp. 1646–1654, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. Springer, 2014.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *ECCV*, pp. 21–37. Springer, 2016.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *NeurIPS*, 33:11525–11538, 2020.

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

Lovett, A. and Forbus, K. Modeling visual problem solving as analogical reasoning. *Psychological review*, 124(1):60, 2017.

Lovett, A., Tomai, E., Forbus, K., and Usher, J. Solving geometric analogy problems through two-stage analogical mapping. *Cognitive science*, 33(7):1192–1231, 2009.

Lovett, A., Forbus, K., and Usher, J. A structure-mapping model of raven's progressive matrices. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

Lu, J., Yang, J., Batra, D., and Parikh, D. Hierarchical question-image co-attention for visual question answering. *NeurIPS*, 29:289–297, 2016.

Mehta, M. R. Neuronal dynamics of predictive coding. *The Neuroscientist*, 7(6):490–495, 2001.

Mondal, S. S., Webb, T. W., and Cohen, J. Learning to reason over visual objects. In *ICLR*, 2022.

Nie, W., Yu, Z., Mao, L., Patel, A. B., Zhu, Y., and Anandkumar, A. Bongard-logo: A new benchmark for human-level concept learning and reasoning. *NeurIPS*, 33:16468–16480, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pang, Z., O'May, C. B., Choksi, B., and VanRullen, R. Predictive coding feedback results in perceived illusory contours in a recurrent neural network. *arXiv preprint arXiv:2102.01955*, 2021.

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

Raven, J. C. and Court, J. *Raven's progressive matrices*. Western Psychological Services Los Angeles, CA, 1938.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. *NeurIPS*, 30, 2017.

Schmidhuber, J. and Heil, S. Predictive coding with neural nets: Application to text compression. In *NeurIPS*, pp. 1047–1054, 1995.

Schmidhuber, J. and Heil, S. Sequential neural text compression. *IEEE Transactions on Neural Networks*, 7(1): 142–146, 1996.

Schultz, W., Dayan, P., and Montague, P. R. A neural substrate of prediction and reward. *Science*, 275(5306): 1593–1599, 1997.

Shekhar, S. and Taylor, G. W. Neural structure mapping for learning abstract visual analogies. 2021.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

Smith, E. C. and Lewicki, M. S. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

Spratley, S., Ehinger, K., and Miller, T. A closer look at generalisation in raven. In *ECCV*, pp. 601–616. Springer, 2020.

Spratling, M. W. Predictive coding as a model of cognition. *Cognitive processing*, 17(3):279–305, 2016.

Srinivasan, M. V., Laughlin, S. B., and Dubs, A. Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205):427–459, 1982.

Sterzer, P., Voss, M., Schlagenhauf, F., and Heinz, A. Decision-making in schizophrenia: A predictive-coding perspective. *NeuroImage*, 190:133–143, 2019.

Summerfield, C., Egner, T., Greene, M., Koechlin, E., Mangels, J., and Hirsch, J. Predictive codes for forthcoming perception in the frontal cortex. *Science*, 314(5803):1311–1314, 2006.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pp. 4489–4497, 2015.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Wang, D., Jamnik, M., and Lio, P. Abstract diagrammatic reasoning with multiplex graph networks. *arXiv preprint arXiv:2006.11197*, 2020.

Wang, K. and Su, Z. Automatic generation of raven's progressive matrices. In *IJCAI*, 2015.

Webb, T., Dulberg, Z., Frankland, S., Petrov, A., O'Reilly, R., and Cohen, J. Learning representations that support extrapolation. In *ICML*, pp. 10136–10146. PMLR, 2020.

Wen, H., Han, K., Shi, J., Zhang, Y., Culurciello, E., and Liu, Z. Deep predictive coding network for object recognition. In *ICML*, pp. 5266–5275. PMLR, 2018.

Wu, Y., Dong, H., Grosse, R., and Ba, J. The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning. *arXiv preprint arXiv:2007.04212*, 2020.

Zhang, C., Gao, F., Jia, B., Zhu, Y., and Zhu, S.-C. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, pp. 5317–5327, 2019a.

Zhang, C., Jia, B., Gao, F., Zhu, Y., Lu, H., and Zhu, S.-C. Learning perceptual inference by contrasting. *NeurIPS*, 2019b.

Zhang, C., Jia, B., Zhu, S.-C., and Zhu, Y. Abstract spatial-temporal reasoning via probabilistic abduction and execution. In *CVPR*, pp. 9736–9746, 2021.

Zhang, C., Xie, S., Jia, B., Wu, Y. N., Zhu, S.-C., and Zhu, Y. Learning algebraic representation for systematic generalization in abstract reasoning. In *ECCV*, pp. 692–709. Springer, 2022.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017a.

Zhang, R. Making convolutional networks shift-invariant again. In *ICML*, pp. 7324–7334. PMLR, 2019.

Zhang, R., Isola, P., and Efros, A. A. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, pp. 1058–1067, 2017b.

Zheng, K., Zha, Z.-J., and Wei, W. Abstract reasoning with distracting features. *NeurIPS*, 32:5842–5853, 2019.

Zhuo, T. and Kankanhalli, M. Effective abstract reasoning with dual-contrast network. In *ICLR*, 2020.