Neural Prediction Errors as a Unified Cue for Abstract Visual Reasoning

Lingxiao Yang, *Member, IEEE*, Xiaohua Xie, *Member, IEEE*, Wei-Shi Zheng, Fang Fang, and Ru-Yuan Zhang

***Abstract*—Humans exhibit remarkable abilities in recognizing relationships and performing complex reasoning. In contrast, deep neural networks have long been critiqued for their limitations in abstract visual reasoning (AVR), a key challenge in achieving artificial general intelligence. Drawing on the well-known concept of prediction errors from neuroscience, we propose that prediction errors can serve as a unified mechanism for both supervised and self-supervised learning in AVR. In our novel supervised learning model, AVR is framed as a prediction-and-matching process, where the central component is the discrepancy (i.e., prediction error) between a predicted feature based on abstract rules and candidate features within a reasoning context. In the self-supervised model, prediction errors as a key component unify the learning and inference processes. Both supervised and self-supervised prediction-based models achieve state-of-the-art performance on a broad range of AVR datasets and task conditions. Most notably, hierarchical prediction errors in the supervised model automatically decrease during training, an emergent phenomenon closely resembling the decrease of dopamine signals observed in biological learning. These findings underscore the critical role of prediction errors in AVR and highlight the potential of leveraging neuroscience theories to advance computational models for high-level cognition in artificial intelligence.**

Received 15 October 2024; revised 19 August 2025; accepted 1 October 2025. Date of publication 20 October 2025; date of current version 9 January 2026. This work was supported by the NSFC Projects under Grant T2421004, Grant 62206316, Grant 32441102, and Grant 31930053, in part by Shanghai Municipal Education Commission under Grant 2024AIZD014, in part by the Project of Guangdong Provincial Key Laboratory of Information Security Technology under Grant 2023B1212060026, and in part by National Science and Technology Innovation 2030 Major Program under Grant 2022ZD0204802. Recommended for acceptance by B. Hammer. *(Corresponding authors: Fang Fang; Ru-Yuan Zhang.)*

Lingxiao Yang is with the School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China (e-mail: yanglx9@mail.sysu.edu.cn).

Xiaohua Xie and Wei-Shi Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China, also with the Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510275, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, MOE, Guangzhou 51000, China.

Fang Fang is with the School of Psychological and Cognitive Sciences and Beijing Key Laboratory of Behavior and Mental Health, IDG/McGovern Institute for Brain Research, Peking-Tsinghua Center for Life Sciences, and Key Laboratory of Machine Perception, MOE, Peking University, Beijing 100871, China (e-mail: ffang@pku.edu.cn).

Ru-Yuan Zhang is with the Brain Health Institute, National Center for Mental Disorders, Shanghai Mental Health Center, School of Medicine and School of Psychology, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ruyuanzhang@sjtu.edu.cn).

The code is available at https://github.com/ZjjConan/AVR-PredRNet.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2025.3623461, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2025.3623461

***Index Terms*—Neural prediction errors, analogical visual reasoning, intelligence tests, RPM-like, bongard problem.**

I. Introduction

With the growing volume of data and model complexity, modern artificial intelligence (AI) models excel in perceptual tasks like text classification [1], image restoration [2], [3], object detection [4], [5], [6], [7], and video comprehension [8], [9]. However, they still struggle with complex reasoning tasks, such as **abstract visual reasoning (AVR)**, which is considered a key milestone toward achieving artificial general intelligence.

The gold standard for testing intelligence in machine learning is the Turing test. However, the language-based Turing test is inappropriate for assessing AVR. In AVR tasks, observers should learn abstract rules within a visual context set and then infer the correct answer based on those rules. A widely used AVR test in psychology is Raven's Progressive Matrices (RPMs) [16]. Fig. 1(a)–(d) show a few RPM-like problems from RAVEN [10], PGM [11], CLEVR-Matrices [12] and VAD [13]. In these tasks, an observer is asked to select the correct answer from provided choice images to fill the missing panel (denoted by the ? marker), where the multiple rows or columns form the same abstract rules (e.g., color, size, progression, etc). Besides RPM-like problems, other visual reasoning tasks, such as Bongard problems [17], are also used to test intelligence. In each Bongard problem, there are six positive context images and six negative context images. The goal is to determine whether the provided choice image follows the same rules as the positive set or not.

Recently, AVR tasks have gained prominence in machine learning. Traditional AVR tasks used in psychology typically include only a small number of problems and object relationships, and are often not publicly available. However, recent benchmark datasets for RPM-like problems have significantly accelerated this line of research [10], [11]. Although some methods have shown strong performance on specific datasets, few studies achieve impressive results across all datasets, especially in scenarios such as in-distribution evaluation, out-of-distribution (OOD) generalization, few-shot learning and reasoning on complex natural images.

Another open question concerns the learning settings. Existing deep models for AVR mostly require large amounts of samples for supervised training. Although these studies enhance our understanding of AVR in deep learning, the supervised learning strategy is fundamentally different from that used by humans. In

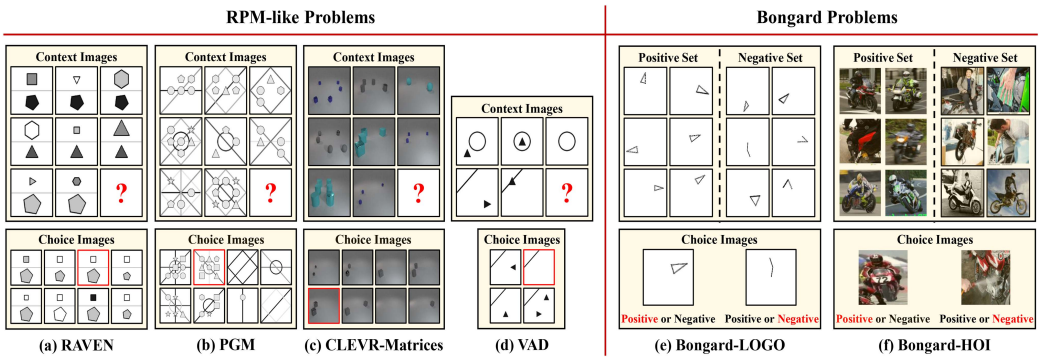


Fig. 1. An illustration of two type of AVR problems used in this study from different datasets: the RPM-like problems from (1) the 16-images-based RAVEN-type problems, including RAVEN [10], PGM [11] and CLEVR-Matrices [12], and (2) the 9-images-based VAD [13] problems, and the Bongard problems from Bongard-LOGO [14] and Bongard-HOI [15] datasets. In each RPM-like problem as shown in (a)–(d), a few context images are provided. The goal is to choose the correct one (highlighted in red) from the provided choice images to fill in the missing one (denoted by **?**), making rows or columns with similar patterns. The Bongard problems in (e) and (f) provide two different sets: one is positive set containing similar patterns, and the other one is negative set with different patterns. Methods should classify which set is belong to for the two provided choice images. Obviously, for all tasks, a subject should recognize diverse visual objects and their attributes, and then discover relationships among these objects for inference.

a standard human intelligence test, a human observer typically engages with different AVR problems directly and performs self-supervised learning for correction. This self-supervised learning strategy, common in humans, has been largely underexplored in recent AVR studies.

How should deep models solve complex AVR problems? A well-known principle in biological learning is minimizing **prediction errors (PEs)**, where primates actively predict incoming sensory signals and adjust based on the error between predictions and actual feedback [18]. Inspired by this idea, we leverage this well-established concept – PEs – as a central mechanism for supervised and self-supervised AVR learning. In supervised learning, we propose the **predictive reasoning block (PRB)**, and frame AVR as a prediction-and-matching process [19], where an agent iteratively generates predictions and calculates the discrepancy (i.e., PEs) between such predictions and the choices. This approach bypasses the need to explicitly learn and represent abstract rules. Built on PRB, we propose a new network called **PredRNet**, which achieves state-of-the-art average performance and superior generalization capabilities across a wide variety of AVR benchmarks. Surprisingly, we observe that PEs in PredRNet automatically decrease during model training, an emergent phenomenon strikingly similar to the reduction in dopamine signals observed during biological learning. For self-supervised learning, we also mimic the minimization of dopamine signals in biological learning by setting PE as an objective in contrastive learning. This method yields a new self-supervised model, **SSPredRNet:** Overall, we emphasize that PE is vital in both supervised and self-supervised AVR.

Partial results of this work have been presented at *ICML'2023* [20]. Beyond the conference version, the present paper makes the following new advances:

- We propose a new self-supervised learning framework for AVR, termed *SSPredRNet*, where predictive error (PE) is exploited as the key signal to construct training samples and support inference in novel reasoning tasks.
- We are, to the best of our knowledge, the first to unify supervised and self-supervised AVR under the neuroscience-inspired principle of PE.
- We conduct extensive comparisons with state-of-the-art self-supervised methods.
- We further evaluate both PredRNet and SSPredRNet on natural image datasets.
- Comprehensive results on **19** benchmark datasets confirm that our models achieve leading performance across diverse machine learning evaluation protocols.

Overall, this study thus highlights PE as a neuroscience-grounded principle for visual reasoning, and provides strong baselines for future work in AVR.

II. Related Work

Here, we first discuss studies on two related types of AVR tasks(e.g., RPM-like and Bongard problems) and then introduce the main idea of PEs in both neuroscience and computer vision.

A. Abstract Visual Reasoning

Reviewing all AVR tasks is beyond the scope of this work. We kindly refer readers to a very recent survey [21] for a more comprehensive analysis. Here, we focus on two classes of tasks: RPM-like problems – the most commonly studied task in recent deep learning research, and Bongard problems – a few-shot visual reasoning task.

RPM-like Problems: As a standard type of IQ test in psychology, RPM-like problems [11], [12], [13], [16] (see Fig. 1(a)–(d)) are useful for understanding human abstract and analogical reasoning abilities [22]. Developing computer algorithms to solve RPM-like problems is a promising area of research in artificial intelligence, particularly in the pursuit of human-level intelligence and beyond. Early studies in psychology typically rely on cognitive theories, such as structure-mapping [23], analogical perception [24], and symbolic representation [22], to heuristically solve RPM-like problems. However, the RPM-like problems used in these studies are manually designed by human experts and thus limited in size. Such small-scale datasets do not provide a comprehensive testbed for modern machine learning and computer vision algorithms.

To accelerate relevant research in machine learning, Wang and Su [25] use first-order logic to formulate RPM-like problems and automatically generate a large number of them. Based on this dataset, Hoshen and Werman [26] propose perhaps the first deep neural network (DNN) to solve simple geometric patterns. Barrett et al. [11] argue that most existing DNNs are unsuitable for discovering high-level abstract statistical relations, leading to inferior performance on RPM-like problems. To address this, they introduce a relation module [27] for convolutional neural networks (CNNs), forming a new network architecture – Wild Relation Network (WReN). Beyond network architecture development, two large-scale RPM-like datasets – RAVEN [10] (Fig. 1(a)) and PGM [11] (Fig. 1(b)) – are also created. Some studies point out the defects in the original RAVEN and propose improvements such as RAVEN-FAIR [28] and Impartial-RAVEN [29]. More recently, Spratley et al. [30] develop an anti-objectivist visual reasoning dataset. These new datasets significantly accelerate the development of novel methods, including those that explore row-wise and column-wise relationships [29], [31], [32], [33], [34], discover multi-scale patterns [28], [35], improve relation modules with MLP [36] or transformer [12], [37], design neuro-symbolic representations [38], [39], [40], use commutative algebra for matrix representation [41], fuse features through graph networks [42], employ multi-label learning [43], and learn models under semi-supervised [44] and self-supervised settings [45], [46].

Bongard problems: Besides the RPM-like problems, several studies propose to address another AVR task — Bongard problems. Over fifty years ago, a collection of one hundred human-designed problems has been invented by M. M. Bongard [17] to demonstrate the large gap between high-level human cognition and computer algorithms. In this task, humans or machine models need to learn high-level of concepts from a few support images, where the concepts are goal-oriented, context-dependent, and analogical [47]. After that, the subjects should use the learned concepts to reason the correct labels of query images, i.e., belonging to positive or negative sets. For this task, a few machine algorithms have been proposed, including casting Bongard problems as an inductive logic programming [48] problem or a concept communication [49] problem. A more recent report [50] casts each Bongard problem as a few-shot learning problem and solve this problem by DNNs. To advance the development of machine models, Nie et al. [14] develop a program-guided generation method to produce a large set of Bongard problems in action-oriented LOGO language (Fig. 1(e)). In addition, another Bongard-like problem – Bongard-HOI [15] (Fig. 1(f)) has been designed for understanding human-object interactions (HOI) in natural images.

B. Prediction Error in Neuroscience

PE is a well-known concept in neuroscience. The conventional view of sensory processing postulates that cortical neurons encode absolute magnitudes of stimulus input. The PE instead emphasizes that our brain maintains an internal model and actively generates predictions for incoming signals. What cortical neurons actually encode are **the relative differences** between the predictions and the actual sensory input, rather than **the absolute sensory input** ***per se*** .

Schultz et al. [51] pioneer the study of reward PE and show 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 is later introduced into sensory processing. Rao and Ballard [18] incorporate PE into a three-layer neural network and find that the neural receptive fields after training show strong similarities to the center-surround effects reported in the neurophysiological literature [52], [53], [54], [55], [56]. The concept of PE has now been extended to the auditory system [57], the hippocampus [58], and the prefrontal cortex [59].

Friston and Kiebel [60] further extend this idea and propose 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 PEs to update the belief held in the internal model. This theory can explain a wide range of cognitive phenomena, including binocular rivalry [61], reinforcement learning [62], visual illusions [63], and even atypical behavior in psychiatric populations [64] *etc*.

In summary, the PE is one of the most fundamental neuroscientific concepts and may make a signiticant contribution to AI when building human-like models or beyond.

C. Prediction Error in Computer Vision

Prediction-based processing has also been introduced into the field of data compression in many years ago [65]. In recent computer vision research, some works have also used prediction-based processing as loss functions or training strategies for various applications [66], [67], [68], [69], [70], [71], [72], [73]. For example, Karol and Yann [68] train a predictor to approximate the original sparse codes to improve inference speed. Carl et al. [67] propose an self-supervised learning method by predicting the relative position of image patches. Zhang et al. [71] design a framework to learn good representations by estimating color images from grayscale images. Oord [72] et al. propose to learn good features by predicting the future in latent space with powerful autoregressive models. More recently, He et al. [74] demonstrate that good representations of deep models can be learned by predicting images from their corrupted ones.

Instead of using PE only as a loss function or a training strategy, a few works [75], [76], [77] incorporate PE into the network architecture for object recognition. Although our work uses PE like several previous studies, we focus on AVR 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 [75], [76], [77] emphasize the computation of PEs across all layers. Third, our model iterates prediction-based processing in a stacked fashion without any

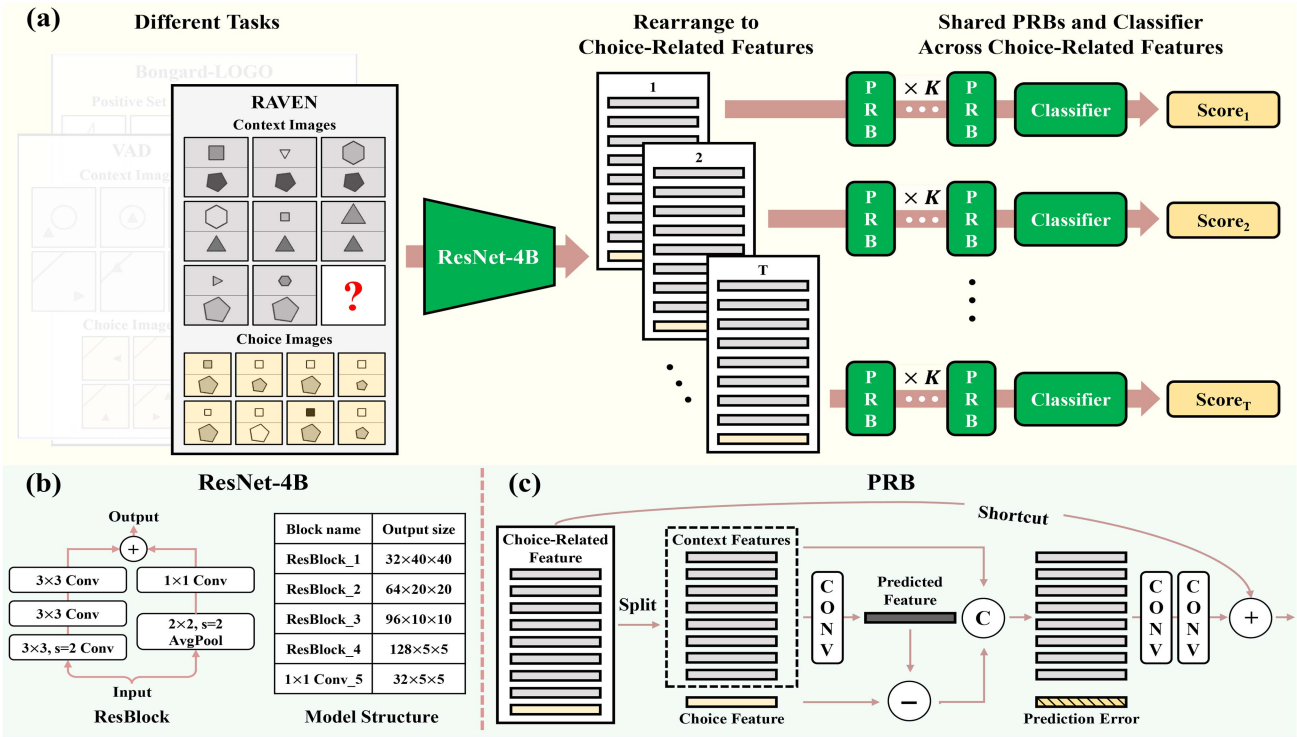


Fig. 2. (a) An overview of the proposed PredRNet. It contains an image encoder – ResNet-4B, to parallel transform input images to features, a stacked structure having multiple predictive reasoning blocks (PRBs) to abstract relationships between features from context images and choice images, and a classifier to obtain score for each choice. (b) Structures of our ResBlock and ResNet-4B. For a 80×80 image, the output size of each block is also shown. (c) Details of the proposed PRB.

recurrent connection. Furthermore, our model can be easily employed in supervised and self-supervised settings without too much engineering adaption.

III. PredRNet

Our PredRNet contains three components: (1) an **Image Encoder** that transforms each image into a 3-dimensional high-level representation (features), (2) multiple ($K \geq 2$) stacked **Predictive Reasoning Blocks (PRB)** to extract relationships between the independent srepresentations of the context and choice images, and (3) a **Classifier** that outputs scores for the choice images.

A. Image Encoder

Several studies have provided baseline results using popular networks like ResNet-18 or ResNet-50 [78] and their extended variants (hereafter referred to as baseline networks) [10], [11], [29], [31], [45], [46]. For example, SRAN [29] combines three ResNet-18 models to extract features and then uses their proposed structure to discover rules. PRD [45] and NCD [46] leverage the ImageNet [79] pre-trained ResNet-18 to build robust feature representations in the self-supervised setting of RAVEN problems. 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, not for the typical smaller images used in AVR problems [80] (Fig. 1). Additionally, some of these baseline networks fuse all images of an AVR problem together at the first layer. This "early fusion" approach only captures low-level relationships between images rather than high-level relationships, limiting reasoning performance. Therefore, we propose a new ResNet variant to provide a stronger baseline for solving AVR problems.

Our image encoder has four **ResBlocks**, each consisting of a residual branch and a shortcut branch. The residual branch contains three convolutional layers. The shortcut branch first applies an average pooling layer to downsample inputs [81] and then matches the output size of the residual branch by a 1×1 convolutional layer. These two branches are then added together to form the input for the next block. After the four blocks, we append a 1×1 convolutional layer ($Conv_5$) to reduce the feature dimension for further processing. The overall structure of our image encoder is shown in **Fig. 2(b) right**. Importantly, all context images (e.g., eight images in RAVEN-type problems) are processed indpendently by the image encoder to create distinct features, without any fusion across context images. More details about the image encoder can be found in Appendix A, available online. We will refer to our image encoder as **ResNet-4B**.

B. Rearranging to Choice-Related Features

Our ResNet-4B extracts features from all context and choice images (e.g., eight context and eight choice images in a RAVEN problem) in parallel, without exploring the relationships among them. However, the core of AVR tasks lies in examining relationships between context and choice features. Therefore, it is

crucial to combine these image features to form **choice-related features**, and an appropriate relational module can be used to extract relationships among them. This rearrangement depends on the exact structure of the problem. We now introduce our rearrangement operation separately for each dataset.

Let $\boldsymbol{x}_i \in \mathcal{R}^{D\times H\times W}$ and $\boldsymbol{t}_j \in \mathcal{R}^{D\times H\times W}$ represent the i-th context and j-th choice features, respectively. C and T are the numbers of context and choice features. The concatenation operator is denoted by $[.,.]$. The choice-related features of different problems are defined as:

- *RAVEN-type problems:* RAVEN-type problems include RAVEN, I-RAVEN, RAVEN-FAIR, PGM, and CLEVR-Matrices datasets. Fig. 1(a)–(c) show examples. These problems contain $C = 8$ context and $T = 8$ choice images. We concatenate eight context features and one choice feature to form a choice-related feature:

$$\boldsymbol{X}_j = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_8, \boldsymbol{t}_j], \quad j = 1, 2, \ldots, 8, \tag{1}$$

  where $\boldsymbol{X}_j \in \mathcal{R}^{9\times D\times H\times W}$ is the j-th choice-related feature.
- *VAD problems:* A VAD example is shown in Fig. 1(d). It has a structure similar to other RPM-like problems but contains only $C = 5$ context images and $T = 4$ choice images. Similar to (1), we can formulate a choice-related feature of VAD by concatenating the five context features with a choice feature as:

$$\boldsymbol{X}_j = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_5, \boldsymbol{t}_j], \quad j = 1, 2, \ldots, 4, \tag{2}$$

  where $\boldsymbol{X}_j \in \mathcal{R}^{6\times D\times H\times W}$ is the j-th choice-related feature.
- *Bongard problems:* Examples of Bongard -LOGO and Bongard-HOI are shown in Fig. 1(e) and (f). Each Bongard problem contains six positive and six negative context images, and two choice images. Following [14], we construct two choice-related features for each choice image feature $\boldsymbol{t}_j, j = 1, 2$ as:

$$\boldsymbol{X}_{[j,1]} = [\boldsymbol{x}_1^{pos}, \boldsymbol{x}_2^{pos}, \ldots, \boldsymbol{x}_6^{pos}, \boldsymbol{t}_j], \tag{3}$$

$$\boldsymbol{X}_{[j,2]} = [\boldsymbol{x}_1^{neg}, \boldsymbol{x}_2^{neg}, \ldots, \boldsymbol{x}_6^{neg}, \boldsymbol{t}_j], \tag{4}$$

  where $\boldsymbol{x}_i^{pos}$ and $\boldsymbol{x}_i^{neg}$ denote a positive and a negative context feature respectively. Each choice-related feature contains $C = 6$ context and $T = 1$ choice features. Thus, $\boldsymbol{X}_{[j,1]}$ and $\boldsymbol{X}_{[j,2]}$ have the same shape of $\mathcal{R}^{7\times D\times H\times W}$.

In summary, each choice-related feature has the same shape formulation: $\boldsymbol{X}_j \in \mathcal{R}^{(C+1)\times D\times H\times W}$, where C indicates the number of context features.

C. Stacked Predictive Reasoning Blocks

Although solving AVR problems is a high-level cognitive task, we argue that it still follows the prediction-and-matching process as in perceptual systems [19]. For example, in a RPM-like problem, an observer must first examine all context images to learn the implicit rule. Based on this learned rule, the observer then predicts what the correct answer should be and match this prediction to the choice images. The matching step can be formulated as calculating the error between the prediction and the choice images, and the PEs are subsequently used to refine the learned rule. Since the initial learned rules and predictions may be incorrect, the prediction-and-matching process should be iterated. Thus, we argue that PE is the critical cue for correct reasoning. Based on this theory, we propose a novel reasoning block – PRB, Fig. 2(c).

The PRB is the most important component of our PredRNet. Its goal is to extract relationships among all features within a choice-related feature, $\boldsymbol{X}_j \in \mathcal{R}^{(C+1)\times D\times H\times W}$. To ease implementation in toolkits like PyTorch, we first transform each choice-related feature $\boldsymbol{X}_j$ from $\mathcal{R}^{(C+1)\times D\times H\times W}$ to $\mathcal{R}^{D\times(C+1)\times L}$, where $L = H \times W$. After this transformation, a built-in PyTorch function – $Conv2D$ – can be applied simultaneously along the 2-nd dimension to abstract context-choice relationships, and along the 3-rd dimension to extract spatial cues. We then split $\boldsymbol{X}_j$ into two feature sets: the C context features $\boldsymbol{X}_c \in \mathcal{R}^{D\times C\times L}$ and a single choice feature $\boldsymbol{t}_j \in \mathcal{R}^{D\times 1\times L}$ (output after **Split** in Fig. 2(c)). A $C \times 1$ convolutional layer with D channels combines all context features into a single predicted feature $\boldsymbol{x}_p \in \mathcal{R}^{D\times 1\times L}$, which has the same size as the choice feature. The difference between the predicted feature and the choice feature constitutes PE. We show the process as:

$$[\boldsymbol{X}_c, \boldsymbol{t}_j] = Split(\boldsymbol{X}_j), \tag{5}$$

$$\boldsymbol{x}_p = BN(Conv_{C\times 1}(\boldsymbol{X}_c)), \tag{6}$$

$$\boldsymbol{e}_j = ReLU(\boldsymbol{t}_j) - ReLU(\boldsymbol{x}_p), \tag{7}$$

where the PE $\boldsymbol{e}_j \in \mathcal{R}^{D\times 1\times L}$ represents the discrepancy between the prediction from the context features and the j-th choice feature.

We then concatenate the PE $\boldsymbol{e}_j$ with the original context features $\boldsymbol{X}_c$ along the 2-nd dimension and pass them through two additional convolutional layers. Similar to our ResBlock, a shortcut is added to facilitate optimization. This process can be formulated as:

$$\boldsymbol{Y}_j^0 = [\boldsymbol{X}_c, \boldsymbol{e}_j], \quad \boldsymbol{Y}_j^0 \in \mathcal{R}^{D\times(C+1)\times L}, \tag{8}$$

$$\boldsymbol{Y}_j^l = ReLU(BN^l(Conv_{3\times 3}^l(\boldsymbol{Y}_j^{l-1}))), \quad l = 1, 2, \tag{9}$$

$$\boldsymbol{Y}'_j = \boldsymbol{Y}_j^2 + BN(Conv_{1\times 1}(\boldsymbol{X}_j)), \tag{10}$$

where $\boldsymbol{Y}'_j$ is the output of this PRB.

By far, our PRB extracts the relationships between context and choice features. However, a single prediction-and-matching is likely insufficient. We design a stacked structure combining several ($K \geq 2$) PRBs to gradually refine the learned rules (Fig. 2(a)). In our implementation, both errors and context features are forwarded for further processing in each PRB (see (8)–(10)).

D. Classifier

Similar to previous models [28], [31], [36], our classifier is an MLP that outputs a single score for the j-th choice-related feature: $score_j = MLP(Flatten(\boldsymbol{Y}'_j))$. For each RPM-like problem, our PredRNet outputs T scores in parallel, i.e., $T = 8$ for RAVEN-type problems, and $T = 4$ for VAD problems. The choice image with the highest score across T scores is selected as the final result. For each choice image of a Bongard problem, we select the result with the highest score between outputs from

(3) and (4). For model training, we use the same binary cross-entropy loss as in [28], [31], [32] for all RPM-like problems, and the default cross-entropy [14], [15] for the Bongard problems.

IV. Self-Supervised PredRNet – SSPredRNet

Current state-of-the-art methods rely heavily on large amounts of labeled data for supervised training. In contrast, humans can learn from vast amounts of unlabeled data in an self-supervised manner, demonstrating strong deductive and inferential abilities. Our goal is to further develop an self-supervised learning model for AVR.

A commonly used strategy for self-supervised learning is contrastive learning. However, applying contrastive learning to abstract rules presents significant challenges for two reasons. First, constructing effective positive and negative samples for contrastive learning is difficult. The same abstract rule can be expressed through different objects, while a single object can participate in the expression of multiple rules. The positive and negative samples need to be defined at the abstract level, rather than merely being differentiated on image-level properties. Second, AVR tasks require the integration of contrastive information into both learning and inference. Specifically, the method used to construct positive and negative samples during learning should also be readily applicable during inference when the model encounters new AVR problems.

We propose that PEs in PredRNet can naturally extend to constructing positive and negative samples for contrastive learning. PEs can be used as a criterion for selecting choices during inference across various tasks, serving as a crucial mechanism that unifies both learning and inference in self-supervised learning.

Building on this concept, we introduce a new self-supervised model – **SSPredRNet**. This approach ensures that all information comes from a single problem, and that samples are genuinely "positive" or "negative" (see Discussion Section VI-B for a comparison with previous methods). The calculation of PEs allows for seamless inference on new problems.

A. Construction of Samples for PRB Reasoning

We construct training samples based on two facts: **Fact (1)** each problem contains complete and correct rules in context images, e.g., the first two rows of context images in RAVEN problems; **Fact (2)** changing any one of these context images has a high probability of breaking the rules. Based on these facts, we can easily construct positive and negative training examples for self-supervised training (see Fig. 3).

Similar to the supervised model, self-supervised learning of AVR also requires rearranging all features extracted by our image encoder in a manner tailored to each specific task for relationship extraction, as shown:

- *RAVEN-type problems:* The positive and negative feature sets are constructed as:

$$\boldsymbol{X}_1^{pos} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_5, \boldsymbol{x}_6], \tag{11}$$

$$\boldsymbol{X}_j^{neg} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_5, \boldsymbol{t}_j], \quad j = 1, 2, \ldots, 8, \tag{12}$$

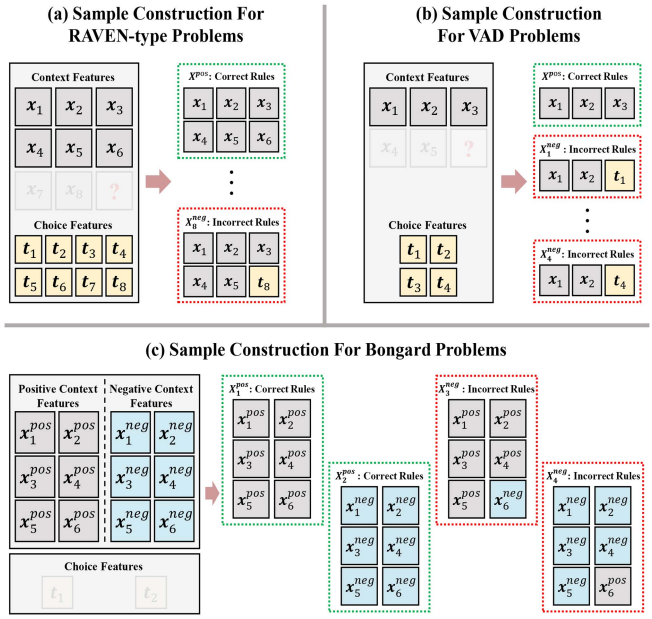


Fig. 3. Different sample construction methods after feature extraction. AVR problems often contain correct rules within their context images, e.g., the first two rows in a RAVEN problem. Therefore, replacing any context image will largely disrupt these rules, resulting in samples with incorrect rules. Since different tasks have different prior structures, we provide different sample construction methods, from (a) to (c). Features colored in lightgrey are not used during the training phase to avoid including correct rules as negative samples for optimization. Although there are slight differences in the construction methods, our central idea remains the same: disrupting contextual information to generate positive and negative errors for contrastive learning.

- *VAD problems:* Similar to all RAVEN-type problems, these feature sets can be formulated as:

$$\boldsymbol{X}_1^{pos} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3], \tag{13}$$

$$\boldsymbol{X}_j^{neg} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{t}_j], \quad j = 1, 2, 3, 4, \tag{14}$$

- *Bongard problems:* For each problem, we construct four feature sets, defined as following:

$$\boldsymbol{X}_1^{pos} = [\boldsymbol{x}_1^{pos}, \boldsymbol{x}_2^{pos}, \ldots, \boldsymbol{x}_5^{pos}, \boldsymbol{x}_6^{pos}], \tag{15}$$

$$\boldsymbol{X}_2^{pos} = [\boldsymbol{x}_1^{neg}, \boldsymbol{x}_2^{neg}, \ldots, \boldsymbol{x}_5^{neg}, \boldsymbol{x}_6^{neg}], \tag{16}$$

$$\boldsymbol{X}_3^{neg} = [\boldsymbol{x}_1^{pos}, \boldsymbol{x}_2^{pos}, \ldots, \boldsymbol{x}_5^{pos}, \boldsymbol{x}_6^{neg}], \tag{17}$$

$$\boldsymbol{X}_4^{neg} = [\boldsymbol{x}_1^{neg}, \boldsymbol{x}_2^{neg}, \ldots, \boldsymbol{x}_5^{neg}, \boldsymbol{x}_6^{pos}], \tag{18}$$

Given the constructed positive/negative samples, each sample has the shape $\boldsymbol{X}_j^{pos|neg} \in \mathcal{R}^{(C'+1)\times D\times H\times W}$, where $C' = 5, 2, 5$ for the RAVEN-type, VAD, and Bongard problems, respectively. Using these feature sets, we set $C = C'$ in (6) to extract relationships via our PRB. For example, in a RAVEN problem, $C = 8$ in (6) for PredRNet is replaced by $C = 5$ in SSPredRNet, while in a VAD problem, C is changed from 5in PredRNet to 2in SSPredRNet.

B. Prediction Errors and Contrastive Loss

In PredRNet, our PRB uses PEs between context and choice features to represent relationships. The philosophy behind PRB follows the prediction-and-matching process [19]. Our self-supervised learning approach also follows this principle but explicitly uses PEs as optimization goals.

Unlike PredRNet, PEs in SSPredRNet include both positive PEs $\boldsymbol{e}_i^{pos}$ and negative PEs $\boldsymbol{e}_j^{neg}$ for contrastive learning. Here we use the same operations to calculate PEs (5)–(7) except that different choice-related features are used. For RAVEN-type problems, $\boldsymbol{e}_i^{pos}$ and $\boldsymbol{e}_j^{neg}$ are calculated based on the postive and negative choice-related features in (11)–(12). For VAD problems, $\boldsymbol{e}_i^{pos}$ and $\boldsymbol{e}_j^{neg}$ are calculated based on the choice-related features in (13)–(14); and for Bongard problems, $\boldsymbol{e}_i^{pos}$ is calculated based on the choice-related features in (15)–(16), while $\boldsymbol{e}_j^{neg}$ is calculated based on choice-related features in (17)–(18). After that, we define the contrastive loss [82], [83]:

$$\mathcal{L}_{CTR} = \frac{1}{2}\sum_{i=1}^{n^{pos}} ||\boldsymbol{e}_i^{pos}||^2 + \sum_{j=1}^{n^{neg}} (max\{0, \tau - ||\boldsymbol{e}_j^{neg}||\})^2, \tag{19}$$

where $||\cdot||$ is the L_2 distance and τ is a margin parameter defining when incorrect rules have a large distance to be ignored by the loss. n^{pos} and n^{neg} are numbers with correct and incorrect rules in each AVR problem respectively. Moreover, similar to multi-head loss employed on different layers [28], we use $\mathcal{L}_{CTR}$ on multiple PRBs for training.

Intuitively, the first term ($||e_i^{pos}||^2$) in (19) penalizes positive PEs that are too large in positive samples (i.e., all context features). The second term ($\sum_{j=1}^{n^{neg}} (max\{0, \tau - ||\boldsymbol{e}_j^{neg}||\})^2$) penalizes PEs that are too small within negative samples (i.e., context features and one disrupt feature). The model can learn meaningful consistency at the rule-level by continuously optimizing (19).

C. Prediction Errors as Inference Signals

For a **RAVEN-type problem**, which contains eight context and eight choice images (Fig. 1(a)–(c)), we construct two sets for the j-th choice image ($j = 1, 2, \ldots, 8$): one combining the first and third rows: $\boldsymbol{X}_{j1} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{t}_j]$, and the other combining the second and third rows: $\boldsymbol{X}_{j2} = [\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_8, \boldsymbol{t}_j]$. The PRBs are used to extract relationships and output PEs for these two sets – $\boldsymbol{e}_{j1}$ and $\boldsymbol{e}_{j2}$, respectively. The final result is selected by finding the minimum PEs among all eight candidate choices as: $\arg\min_{j\in\{1,\ldots,8\}}(||\boldsymbol{e}_{j1}|| + ||\boldsymbol{e}_{j2}||)$.

For a **VAD problem**, which contains five context and four choice images (Fig. 1(d)), we directly use the second row $\boldsymbol{X}_j = [\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{t}_j]$ $(j = 1, 2, 3, 4)$ to extract PEs. The final result is selected by the minimum PEs among all four candidate choices as: $\arg\min_{j\in\{1,\ldots,4\}}(||\boldsymbol{e}_j||)$.

For a **Bongard problem**, which contains two choice images, six positive context images, and six negative context images, we construct two feature sets for each choice feature: $\boldsymbol{X}_{j1} = [\boldsymbol{x}_1^{pos}, \boldsymbol{x}_2^{pos}, \boldsymbol{x}_3^{pos}, \boldsymbol{x}_4^{pos}, \boldsymbol{x}_5^{pos}, \boldsymbol{t}_j]$, and $\boldsymbol{X}_{j2} = [\boldsymbol{x}_1^{neg}, \boldsymbol{x}_2^{neg}, \boldsymbol{x}_3^{neg}, \boldsymbol{x}_4^{neg}, \boldsymbol{x}_5^{neg}, \boldsymbol{t}_j]$, where $j = 1, 2$. Our PRB output two error sets $\boldsymbol{e}_{j1}$ and $\boldsymbol{e}_{j2}$ for these two feature sets respectively. The label of the j-th choice image is determined by: $\arg\min_{z\in\{1,2\}}(||\boldsymbol{e}_{jz}||)$.

V. Experiments

We first introduce the datasets used in this study and the reasons for selecting these datasets in Section V-A. Implementations of our PredRNet and SSPredRNet are shown in Section V-B. In Sections V-C, V-D, V-E, V-F, and V-G, we present comparisons of our PredRNet and SSPredRNet with many state-of-the-art methods, followed by a number of ablation studies and analysis to better understand our proposed methods in Sections V-H and V-I.

A. Datasets

To demonstrate the effectiveness of our methods, we perform extensive experiments on a broad range of datasets and visual reasoning conditions. These datasets are used for multiple mainstream testing setups in current machine learning, including **in-distribution evaluation** (RAVEN/RAVEN-FAIR/I-RAVEN/PGM-Neutral), **(OOD)** (seven PGM datasets), **visual reasoning in 3D scenes** (CLEVR-matrices), **visual reasoning with limited context information** (five VAD datasets), and **few-shot reasoning** in both symbolic (Bongard-Logo) and natural (Bongard-HOI) images. We emphasize that several datasets, including the seven OOD subdatasets from PGM, five VAD datasets, the novel abstract shape split in Bongard-LOGO, and the natural-image-based Bongard-HOI, have been specifically designed to test a model's ability for compositional generalization. For instance, Bongard-HOI tests the model's ability to recombine learned action-object relations (i.e., learn "straddle bicycle" and "wash car" and test "wash bicycle"). In summary, we evaluate our PredRNet and SSPredRNet models across a total of 19 datasets. To our best knowledge, our evaluations cover the broadest AVR datasets to date. We expect these extensive experiments to highlight the flexibility and scalability of our models. We briefly introduce all the datasets as follows.

Three RAVEN datasets: The original **RAVEN** dataset [10] introduces a 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 problems, yielding a total of 70,000 problems with 1,112,000 images. **RAVEN-FAIR** [28] and **I-RAVEN** [29] are recently developed to fix the shortcut bias in the original RAVEN. Both contain the same context images as RAVEN but differ in the way to generate negative answers. Studies in [28], [29] show that both datasets have no shortcut bias, and thus are better than the original RAVEN in evaluating models. We thus include all three RAVEN datasets.

Eight PGM datasets: The whole **PGM** [11] contains eight different datasets. PGM-Neutral is used for **in-distribution** testing, and the other seven datasets are constructed by leaving some attributes or some rules out of training sets, allowing to examine the OOD generalization of networks. Each dataset has

1,222,000 problems with 119,552,000 images. PGM includes diverse abstract rules (i.e., XOR, OR, Progression and AND) among objects, which are very different from those in all RAVEN datasets.

CLEVR-Matrices: The **CLEVR-Matrices** dataset [12] is another RPM-like dataset based on the widely used visual question answer dataset – CLEVR [84]. This dataset includes 3 configurations, i.e., Logic, Location, and Count. Each configuration has 20,000 problems, including 16,000 for training, 2,000 for validation, and 2,000 for testing. This dataset has three kinds of visual attributes, i.e., shape, size, and color. Rules are independently sampled from the set of {null, constant, distribution-of-3}. To avoid the bias problems in RAVEN [10], the choice images are generated with the same algorithm in [29]. We summarize three RAVEN-like datasets, Eight PGM datasets, and CLEVR-Matrices as RAVEN-type problems because they all contain eight context and eight choice images in each problem.

Five VAD datasets: VAD [13] contains five datasets, each containing around 600,000 problems. As shown in Fig. 1(d), the VAD problems are similar to RAVEN-type problems, but only have five context and four choice images. VAD is designed for evaluating visual reasoning with very limited context information. Because VAD and RAVEN-type problems follow the same format of choosing the correct choice image by inferring abstract rules from context images, we summarize three RAVEN datasets, eight PGM datases, CLEVR-Matrices, and five VAD datasets as RPM-like problems (Fig. 1).

Two Bongard datasets: Bongard-LOGO [14] is constructed for evaluating human-level visual reasoning. Problems in this dataset mimic the Bongard problems [17] where observers infer visual concepts from their contexts with a few examples. Therefore, Bongard-LOGO transforms 12,000 problems into few-shot binary classification tasks with context images for abstracting rules, and choice images for answering. Bongard-HOI [15] is another new visual reasoning benchmark that focuses on compositional learning of human-object interactions from natural images. This dataset has disjoint training, validation, and testing sets. The training set contains 118 concepts and 21,956 few-shot instances. The validation and testing sets have 17,184 and 13,941 few-shot instances respectively, corresponding to 167 and 166 visual concepts.

B. Implementations

We set the filters to [32,64,96,128,32] in our ResNet-4B. The first four numbers are our feature channels in ResBlocks, while the last one is the reduced feature channels before our PRB extraction. In PRB, the filter channels of $Conv$ in (6) is 32 for both PredRNet and SSPredRNet. The filters of the other two $Conv$s are set to 128 and 32, respectively. In addition, we add $K = 3$ PRBs after our image encoder because this gives the best overall performance on the validation sets. The effect of K will be discussed later.

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 is done by the Adam solver [88] with a learning rate of 1e-3, a batch size of 128 for three RAVEN datasets, and 256 for all PGM and VAD datasets. 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 information (e.g., metadata) during training. In addition, for each RAVEN, we report the median result from 3 different trials. For VAD and Bongard datasets, we report average results over 3 trials as suggested in [13], [15]. For all PGM datasets, we only report a single result, similar to many previous works [28], [31], [84]. For SSPredRNet, we fix the learning rate and weight decay as 1e-3 and 1e-5 respectively, and set margin τ in (19) as 0.7.

C. In-Distribution Abstract Visual Reasoning

We first evaluate our methods on RAVEN/RAVEN-FAIR/I-RAVEN/PGM-Neutral for in-distrbution visual reasoning.

PredRNet: In the supervised setting, we compare our PredRNet with several state-of-the-art models, including WReN [11], LEN [33], CoPINet [31], SRAN [29], DCNet [32], SCL [85], MLRN [35], MXGNet [42], Rel-Base [36], MRNet [28], STSN [12] and DRNet [86]. In addition, most previous studies only evaluate their methods on one or two datasets. For a more comprehensive comparison, we directly run their published codes on all these four datasets.

Table I(a) shows all results obtained by supervised models. We obtain three main conclusions. First, our PredRNet and the recently proposed DRNet achieve similar average performance on the four datasets, significantly better than other models. Specifically, DRNet uses a CNN stream and a ViT stream for image-wise feature extraction and integration. After that, a relation module similar to Rel-Base [36] is adopted to extract rules for prediction. Our PredRNet, however, uses PEs for relationships. Thanks to this powerful design, our PredRNet provides strong performance with a simple image encoder, which has significantly smaller number of parameters than DRNet (1.27 M v.s. 24.6 M). Second, some recently proposed methods, such as MLRN, DCNet, and CoPINet, only show good results on one or two benchmark datasets. For example, MLRN obtains nearly perfect result (98%) on PGM, but performs poorly on three RAVEN datasets (all $<$30%). Both DCNet and CoPINet achieve 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 $>$96%). These results demonstrate the robustness of PredRNet in discovering diverse types of rules in different datasets. Third, STSN, SCL, Rel-Base, and MRNet are four competitive models, although they do not perform well as ours. STSN, SCL, and Rel-Base directly extract relations in all eight context images. MRNet deliberately includes row and column relation modules.

SSPredRNet: In the self-supervised setting, we compare our SSPredRNet with two state-of-the-art self-supervised models – NCD [46] and PRD [45]. Table I(b) shows all the results. We have two concluding remarks. First, our SSPredRNet achieves the best performance on all four datasets. Specifically, NCD and PRD obtain similar performance (around 47%) on average. Our SSPredRNet provides a relative performance improvement of averaged 33.4% compared to NCD and PRD. This is possible because our self-supervised method better utilizes the context

TABLE I
ACCURACY (%) ON **FOUR MAIN IN-DISTRIBUTION DATASETS**, INCLUDING PGM NEUTRAL (PGM-N), ORIGINAL RAVEN (RVN-O), RAVEN-FAIR (RVN-F), AND IMPARTIAL-RAVEN (I-RVN). FOR RAVENS, ACCURACY IS OBTAINED BY AVERAGING ACROSS ALL SEVEN CONFIGURATIONS. † INDICATES THE PERFORMANCE WAS NOT REPORTED IN ORIGINAL PAPER, AND IS OBTAINED BY THEIR PUBLISHED CODE. THE BEST AND THE SECOND BEST ON EACH DATASET ARE HIGHLIGHTED BY **BOLD** AND UNDERLINE, RESPECTIVELY.

(a) **Supervised models**

	WReN [11]	LEN [33]	CoPINet [31]	SRAN [29]	DCNet [32]	MLRN [35]	SCL [85]	MXNet [42]	Rel-Base [36]	MRNet [28]	STSN [12]	DRNet [86]	PredRNet Ours
PGM-N	62.6	68.1	56.4	71.3	68.6	98.0	88.9	66.7	85.5	94.5	98.2	**99.1**	**99.1**
RVN-O	16.8	72.9	91.4	54.3†	93.6	12.3†	91.6	83.9	91.7	96.6	89.7†	**96.9**	96.7
RVN-F	30.3	51.0	50.6	72.9†	56.1†	29.5†	90.1†	35.1†	93.5†	88.4	95.4†	97.6	**97.9**
I-RVN	23.8	41.4	46.1	60.8	47.2†	12.3†	95.0	26.8†	91.1†	83.5†	95.7	97.1	**97.2**
Avg	33.4	58.4	61.1	64.8	66.4	38.0	91.4	53.1	90.5	90.8	94.8	**97.7**	**97.7**

(b) **Self-supervised models**

	NCD [46]	PRD [45]	SSPredRNet Ours
PGM-N	47.6	36.4†	**64.9**
RVN-O	37.0	37.9	**57.1**
RVN-F	55.1†	56.5†	**66.5**
I-RVN	48.2	55.9	**62.3**
Avg	47.0	46.7	**62.7**

TABLE II
ACCURACY (%) ON ALL EIGHT PGM DATASETS (1 NEUTRAL AND 7 OOD VERSIONS). H.O: HELD-OUT. THE BEST AND THE SECOND BEST ARE HIGHLIGHTED USING **BOLD** AND UNDERLINE.

(a) **Supervised models**

	WReN [11]	MXNet [42]	MRNet [28]	ARII [34]	DRNet [86]	PredRNet Ours
Neutral	62.6	66.7	93.4	88.0	**99.1**	**99.1**
Interpolation	64.4	65.4	68.1	72.0	**83.8**	75.3
Extrapolation	17.2	18.9	19.2	**29.0**	22.2	22.9
H.O Pairs	27.2	33.6	38.4	50.0	93.7	**99.3**
H.O TriplePairs	41.9	43.3	55.3	64.1	78.1	**98.9**
H.O Triples	19.0	19.9	25.9	32.1	**48.8**	41.2
H.O LineType	14.4	16.7	30.1	16.0	27.9	**33.4**
H.O ShapeColor	12.5	16.6	**16.9**	12.7	13.1	13.1
Avg	32.4	35.1	43.4	45.4	58.3	**60.4**

(b) **Self-supervised models**

	NCD [46]	PRD [45]	SSPredRNet Ours
Neutral	47.6	36.4†	**64.9**
Interpolation	47.0	34.5†	**51.8**
Extrapolation	**24.9**	12.7†	17.0
H.O Pairs	30.2†	26.6†	**32.3**
H.O TriplePairs	32.5†	26.1†	**41.9**
H.O Triples	13.9†	19.6†	**23.7**
H.O LineType	18.7†	14.7†	**20.4**
H.O ShapeColor	**14.7**†	13.6†	12.7
Avg	28.7	23.0	**33.1**

and choices images to generate more reliable training samples, resulting in a better model. Second, comparing Table I(a) and (b), our SSPredRNet, as a self-supervised model, even performs better or is comparable with some recent supervised methods, such as WReN, LEN, CoPINet, SRAN, DCNet, MLRN and MXNet. Some of these works even use additional meta-targets as auxiliary training (e.g., WReN, MXNet, CoPINet).

D. OOD Generalization of Abstract Visual Reasoning

One hallmark of AVR is to apply learned abstract rules to OOD scenarios. We further evaluate the OOD generalization capability of our PredRNet and SSPredRNet.

PredRNet: In the supervised setting, we compare our PredRNet with several existing models including WReN [11], MXNet [42], MRNet [28], ARII [34], and DRNet [86]. Slot Abstractor [37] uses transformers to extract object-centric representations and relationships. However, Slot Abstractor is first trained on PGM-Neutral in an self-supervised manner, and then fine-tuned on other PGM datasets in a supervised manner. We therefore do not include Slot Abstractor. To be more comprehensive, we also include results on PGM Neutral. Table II(a) show all OOD results in the supervised setting. PredRNet obtains the best average OOD generalization results across all PGM OOD datasets. In particular, our PredRNet achieves an average performance of 60.4% , outperforming the second performer DRNet by an absolute gain of $+2.1\%$. our PredRNet demonstrates superior OOD generalization capability, further validating the effectiveness of PEs in RPM-like problems.

SSPredRNet: Again, we compare SSPredRNet against NCD and PRD. Table II(b) shows all OOD results in self-supervised settings. Our SSPredRNet also obtains the best average OOD generalization results in all PGM OOD datasets. SSPredRNet exhibits promising performance and achieves an absolute gain of $+4.4\%$ than the second best method – NCD. In addition, our SSPredRNet performs favorably against two supervised methods – WReN and MXGNet in most of the datasets. Compared to WReN and MXGNet, our SSPredRNet obtains better accuracy on H.O Triples and H.O LineType, and achieves a similar performance on Neutral, Extrapolation and H.O Pairs. All results suggests that SSPredRNet have good OOD generalization capabilities.

E. Abstract Visual Reasoning in 3D Scenes

Previous evaluations only include the RAVEN-like problems with symbolic 2D images (e.q., squares, triangles). However, AVR is a central ability that should be performed on more realistic objects. We thus further evaluate our models on CLEVR-Matrices [12], which is designed for AVR in 3D objects and scenes. The CLEVR-Matrices dataset is derived from the widely used visual question answer dataset – CLEVR [84], which is also a 3D-rendered dataset. As shown in Fig. 1(c), each problem also has eight context and eight choice images. This dataset focuses

TABLE III
ACCURACY (%) ON OTHER RPM-LIKE DATASETS, INCLUDING 3D SHAPE RENDERED PROBLEM – CLEVR-MATRICES [12] AND VAD [13]. THE BEST AND THE SECOND BEST ARE HIGHLIGHTED USING **BOLD** AND UNDERLINE.

(a) **CLEVR-Matrices** [12] **(Chance 12.5%)**

	Supervised				Self-supervised
	MLRN [35]	SCL [85]	STSN [12]	PredRNet Ours	SSPredRNet Ours
Logic	47.4	80.9	99.2	**100.0**	66.2
Location	21.4	65.8	**100.0**	99.5	39.4
Count	23.6	64.9	99.6	**99.9**	43.5
Avg	30.8	70.5	99.6	**99.8**	49.7

(b) **VAD** [13] **(Chance 25%)**

	Supervised			Self-supervised
	LBC [13]	NSM [87]	PredRNet Ours	SSPredRNet Ours
Extrapolation	62±2.0	**74**	72±6.0	35±1.1
Interpolation	93±0.4	93	**97±0.2**	36±1.4
N.D Transfer	87±0.5	88	**96±0.3**	39±1.3
N.D ShapeColor	78±0.4	78	**80±1.0**	35±0.8
N.D LineType	76±2.0	79	**82±1.0**	37±1.1
Avg	79	82	**85**	36

more on the rendered 3D shapes in a scene, with very different object attributes and rules, such as shapes and colors.

PredRNet: We follow the original paper [12] and directly train our PredRNet on all three configurations of CLEVR-Matrices. The performance of PredRNet is compared against to MLRN [35], SCL [85], and STSN [12]. The results are shown in Table III(a). Our **PredRNet** performs very competitively with the attention-based method – STSN [12].

SSPredRNet: To our best knowledge, no self-supervised methods have been evaluated on CLEVR-matrices. We thus directly show the performance of our SSPredRNet in Table III(a). Our SSPredRNet performs significantly better than chance 12.5% and even better than the supervised model – MLRN on this dataset. These results demonstrate our models are also good reasoning solvers for different types of context images and objects.

F. Limited Context Based Abstract Visual Reasoning

RAVEN, PGM, and CLEVR-matrices problems contain eight context and eight choice images, providing rich information for reasoning. However, AVR should be robust if the available information is scarce. We thus further test our models on five VAD datasets [13]. As shown in Fig. 1(d), VAD problems are very similar to RAVEN-type problems, but a VAD problem only contains five context images and four choice images. The limited context information considerably increases the difficulty in reasoning.

PredRNet: We compare our PredRNet against two state-of-the-art models on VAD – LBC [13] and NSM [87]. Table III(b) demonstrates that PredRNet obtains the best performance in four out of five VAD datasets and the second best in the other VAD dataset.

SSPredRNet: Again, no self-supervised models have been evaluated on VAD. We thus directly show the performance of our SSPredRNet in Table III(b). Our SSPredRNet performs significantly better than the chance level (25%), indicating that it indeed learns the abstract relationship given limited context information. The result can also serve as a baseline for future self-supervised AVR models.

G. Few-Shot Abstract Visual Reasoning

Few-shot reasoning is fundamental and arguly most chanllenging in artificial intelligence, particularly for tasks requiring adaptability and generalization from limited data. It often requires models to learn from a few data while applying reasoning to make informed decisions in novel scenarios. Evaluating on these few-shot reasoning problems is the key to examine the versatility, scalability, and robustness of existing computing models. To this end, we evaluate our models on two few-shot reasoning benchmarks – **Bongard-LOGO** [14] based on synthetic images and **Bongard-HOI** [15] based on natural images. Both of them are thought to reflect more human-like reasoning processes and designed to solve 2-way 6-shot classification problem.

PredRNet: We compare our PredRNet against Base-SC, Base-MoCo, ProtoNet on Bongard-LOGO because the three models exhibit the best performance in the original paper [14]. Table IV(a) shows all results on Bongard-LOGO. Our PredRNet obtains the highest accuracy (on average 76.9%) in all four test sets of Bongard-LOGO.

We also compare our PredRNet against WReN and Meta-Base on Bongard-HOI because the two models exhibit the best performance in the original paper [15]. Table IV(b) shows the results on Bongard-HOI. Our original PredRNet (labeled as R-4B) is trained from scratch only on this dataset, and its performance is comparable to Meta-Base (labeled as IN-R50). However, Meta-Base (IN-R50) utilizes the ResNet-50 pretained on ImageNet as its image encoder and has a significantly larger number of parameters than PredRNet (R-4b) (24 M v.s. 1.2 M). We thus replace our image encoder ResNet-4B with the pre-trained image encoder of Meta-Base to form a new version of PredRNet (labeled as IN-R50). PredRNet (IN-R50) achieves the best performance on all four test sets of Bongard-HOI. This apple-to-apple comparison to other methods indicates that our PRB can be flexibly combined with other image encoder and is also effective to extract relationships on natural images.

SSPredRNet: We train our SSPredRNet on both Bongard-LOGO and Bongard-HOI. Due to a lack of comparison models, we again directly show the performance of SSPredRNet alongside of supervised models in Tables IV(a) and (b). We find that the performance of SSPredRNet is significantly above chance and even close to supervised methods. For example, our SSPredRNet exhibits averaged accuracy of 72.6% , much higher than the three existing supervised methods on the Bongard-LOGO dataset. These results strongly support the robustness of SSPredRNet.

TABLE IV
ACCURACY (%) ON TWO FEW-SHOT REASONING BENCHMARKS, INCLUDING BONGARD-LOGO BASED ON SYNTHETIC IMAGES [14] AND BONGARD-HOI [15] BASED ON NATURAL IMAGES. THE BEST AND THE SECOND BEST ARE HIGHLIGHTED USING **BOLD** AND UNDERLINE.

(a) **LOGO** [14] (**Chance 50%**). Test sets are FF: free-form shape, BA: basic shape, CM: combinatorial abstract shape, NV: novel abstract shape. "-" indicates not available due to self-supervised training without using ground-truth labels.

	Supervised				Self-supervised
	Base-SC [14]	Base-MoCo [14]	ProtoNet [14]	PredRNet Ours	SSPredRNet Ours
Train.Acc	75.4±1.0	81.2±0.1	73.3±0.2	**86.0±0.3**	-
FF	66.3±0.6	65.9±1.4	64.6±0.9	**76.6±0.1**	72.9±1.2
BA	73.3±1.3	72.2±0.8	72.4±0.8	**92.9±0.2**	89.2±1.6
CM	63.5±0.3	63.9±0.8	62.4±1.3	**69.0±0.9**	61.4±2.2
NV	63.9±0.8	64.7±0.3	65.4±1.2	**69.0±1.1**	66.7±1.0
Avg	66.8	66.7	66.2	**76.9**	72.6

(b) **HOI** [15] (**Chance 50%**). SA: seen action, SO: seen object, USA: unseen action, USO: unseen object. IN-R50: ImageNet-pretrained ResNet-50. R-4B: our image encoder. We also show results of IN-50 with our PRB in penultimate column.

	Supervised				Self-supervised
	WReN [15]	Meta-Base [15]	PredRNet Ours	PredRNet Ours	SSPredRNet Ours
Image.Enc	IN-R50	IN-R50	R-4B	IN-R50	R-4B
SA-SO	50.31	56.45	57.38	**66.61**	52.53
SA-USO	49.72	56.02	53.19	**63.31**	53.39
USA-SO	49.97	55.60	59.39	**66.30**	58.64
USA-USO	49.01	55.21	53.03	**64.17**	51.61
Avg	49.75	55.82	55.75	**65.10**	54.04

TABLE V
COMPARING DIFFERENT OPERATIONS IN PRB FOR RELATIONSHIP EXTRACTION ON THREE RAVEN DATASETS

	MLP	Rmv.PE	Fwd.Err	Cx/P	Cx/P/C	Fus.+	our PRB
RVN-O	83.2	90.5	94.1	90.1	95.6	95.9	**96.7**
RVN-F	88.1	96.8	96.5	87.7	97.8	97.0	**97.9**
I-RVN	76.0	94.3	92.9	68.0	95.8	95.7	**97.2**
Avg	85.8	93.8	94.5	81.9	96.4	96.2	**97.3**

The comparable performance of SSPredRNet with other supervised models indicates that our self-supervised learning based approach is better suited for Bongard problems as compared to other RPM-like datasets. First, the Bongard dataset is a binary classification problem, where each choice sample is determined as positive or negative. This leads in a random guessing probability of 50% , significantly higher than 12.5% (8-way) in the RAVEN-type datasets and 25% (4-way) in the VAD dataset. Second, we eliminate the third and second row data in RAVEN-type and VAD problems during training to minimize noise. However, this approach simplifies the assessment of 3-row and 2-row consistency to 2-row and 1-row consistency in all RAVEN-type and VAD problems, greatly reducing available samples for rule induction and reasoning. This partial testing fails to fully exploit the complete rule consistency within an RPM-like matrix. While the Bongard problem is unique in that it involves deriving rules from the context of 12 images (i.e., 6 positive and 6 negative samples). Our self-supervised learning based approach interchanges one image from the positive set and one from the negative set. In this case, most context images (i.e., 10 out of 12) are retained during training and testing, making the performance of self-supervised learning close to the performance of supervised learning.

H. Ablation Experiments

We conduct a number of ablation experiments to further evaluate the idea of PE in AVR. All experiment settings are the same as presented in Section V-B.

The number $\boldsymbol{K}$ of PRBs (Fig. 4): 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 the number of PRBs on reasoning performance. We also compare different numbers of PRBs with the case in which additional ResBlocks are appended after image encoder.

We conduct this experiment on four in-distribution datasets – three RAVEN datasets and the PGM-Neutral dataset, and two OOD PGM datasets – PGM-Interpolation and PGM-Extrapolation. All results are shown in Fig. 4. We obtain better performance by increasing PRBs from $K = 1$ to $K = 3$. But adding ResBlocks has no significant effects on performance. We speculate that in the ResNet-4B baseline, the image encoder processes the 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 calculating PEs. This is evident by the fact that adding a single PRB to the ResNet-4B baseline improves performance by about 20% for RAVEN-FAIR and about 50% for PGM-Neutral. Adding 2 or 3 PRBs further improves performance on all datasets, giving new state-of-the-art results in both in-distribution and OOD testing. However, adding PRBs more than 3 degrades the performance in most cases. This phenomenon may be due to overfitting.

The effects of various operations in PRB: PRB is the core component in our models to extract abstract rules. PRB *per se* involves several mathematical operations (see (5)–(10)). Here, we conduct six different ablation studies on PredRNet to address the effects of these operations and designs on reasoning performance. For simplicty, we only include three RAVEN datasets for ablation studies.

1) *MLP:* To directly test the effects of PRBs, we replace our PRBs with MLPs
2) *Rmv.PE:* PE is the central concept and key feature for successful reasoning in our model. We remove the step of computing PEs ((5)–(8)), and replace $\boldsymbol{Y}_j$ with $\boldsymbol{X}_j$ in (9), such that our PRB contains only the two last convolutional layers to extract relationships.
3) *Fwd.Err:* We forward only errors without concatenating errors with eight context features.
4) *Cx/P:* We concatenate and forward eight context features and predicted choice features without calculating PEs.

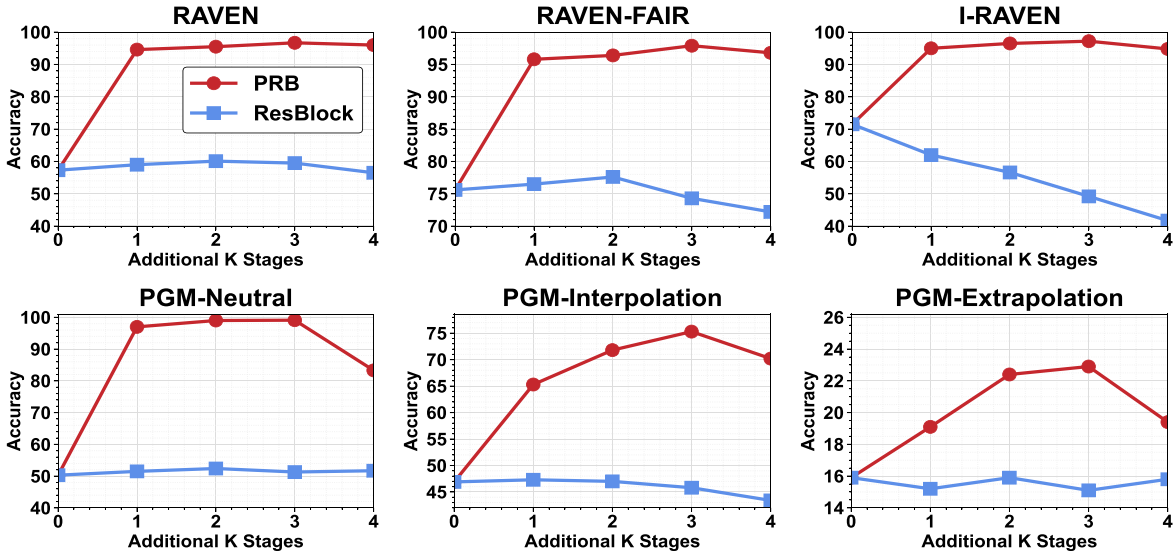


Fig. 4. Analysis of additional K ResBlocks and our PRBs on **four in-distribution** datasets – RAVEN, RAVEN-FAIR, I-RAVEN and PGM-Neutral, as well as two widely used OOD datasets – PGM-Interpolation and PGM-Extraplation. The image encoder in this experiment is our ResNet-4B. $K = 0$ indicates that only the image encoder combined with the classifier, without any relation modules, is used for prediction.

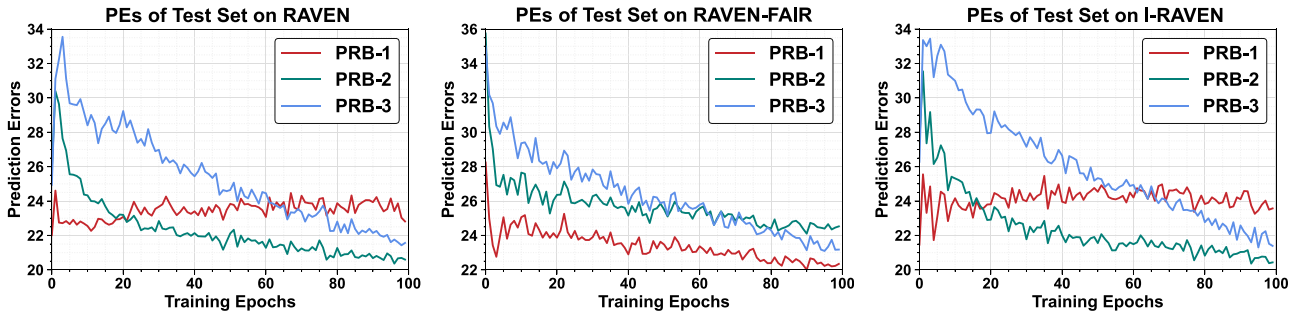


Fig. 5. PEs calculated by (7) on the test sets of all RAVEN datasets along the training progression. Overall, our PredRNet indeed minimizes these errors during the training process in most cases. Best viewed in PDF.

5) *Cx/P/C:* We concatenate and forward eight context features, one predicted feature, and one choice feature without calculating PEs.
6) *Fus.* $+$. The success of our models can also be explained by attention mechanisms, which predict that addition and subtraction make no difference in performance. We therefore replace (7) with $ReLU(x) + ReLU(y)$ like an attention operator. In this case, $(+)$ and $(-)$ have completely different mathematical effects.

We find that removing the key components of PRB or PRB *per se* impairs model performance in all above ablation studies, highlighting the beneficial contributions of the PE calculation and other mathematical operations in PRB.

More ablation studies: In appendix, we include more ablation studies, including the comparisons of different types of image encoders, the representations of PEs, the effects of minimizing PEs explicitly in PredRNet, explicit representation of rules in our PRBs. All these ablation studies suggest that our original design of PRB is the best option.

I. Emergent Minimization of Prediction Errors

The key concept of PEs in neuroscience is that biological agents learn tasks by minimizing these errors. Similarly, in our PredRNet, we explicitly calculate and process PEs, although they are not directly included in the loss function. It remains an open question whether PredRNet learns to reason by minimizing internal PEs, similar to biological learning. To explore this, we computed the PEs for the three PRBs on the test set during model training (see Fig. 5). Surprisingly, we observed that the overall magnitude of PEs significantly decreases as training progresses in most cases. The initial increase of PEs in PRB-1 on RAVEN and I-RAVEN may be due to the unstable representation of relationships in the very early stages of optimization. These findings suggest that our network indeed learns to minimize PEs.

Furthermore, once learning is complete, stacked PRBs are required to simulate information processing from lower to higher-level neurons when a trained model encounters a new problem.

This process continuously uses prediction-and-matching to extract and refine PEs specific to the problem, which is crucial for identifying the correct answer during inference. It is important to note that in SSPredRNet, PEs are explicitly included in the loss to be minimized, so we did not conduct the same analysis for SSPredRNet.

The striking similarity between the behavior of our model and biological brains strongly supports the biological plausibility of both PredRNet and SSPredRNet.

VI. Conclusion and Discussion

In this work, we treat the concept of PE from neuroscience as a central cue in both supervised and self-supervised learning for AVR tasks. Our supervised models (i.e., PredRNet) and self-supervised models (i.e., SSPredRNet) have been extensively tested across various datasets and task regimes, including in-distribution reasoning, OOD generalization, 2D and 3D images, and few-shot reasoning. These results strongly support our models as general and state-of-the-art approaches for AVR. Moreover, the emergent minimization of PE during model training highlights a unique connection between biological and machine learning systems.

A. Supervised and Self-Supervised AVR Models

Though solving the same tasks, PredRNet and SSPredRNet develop different representations of object attributes and abstract rules. This difference primarily stems from the availability of fully labeled samples in supervised learning, which allows the model to learn complete information for each task (e.g., RAVEN, VAD, and Bongard) and make well-informed decisions. In supervised learning, the model can be seen as learning to assess the consistency across three rows or three columns. In contrast, self-supervised learning lacks labeled information. For example, in RAVEN, SSPredRNet learns the relationships between target objects through $[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_6]$, and during testing, it only evaluates the relationships between two rows. In the case of VAD, the available information is even more limited, where partial predictions are made using only the second row.

Our self-supervised AVR model occupies a middle ground between unsupervised and supervised learning. Like unsupervised methods, it does not rely on human-annotated labels; instead, it leverages pretext tasks to generate supervisory signals directly from the data. Such self-supervised approaches have gained substantial traction in both vision and language research, as they combine the label-free nature of unsupervised learning with the effectiveness and scalability needed for large-scale representation learning.

B. Comparison Our SSPredRNet With Other Methods

A common approach in self-supervised learning is to construct positive and negative samples for contrastive learning. However, constructing positive/negative samples based on abstract rules presents a significant challenge. NCD [46] addresses this by using the first two rows of a RAVEN problem as positive samples, while constructing eight negative samples by filling the missing position in the third row with eight choice images. NCD [46] also replaces the last four choice images with images from other RAVEN problems. However, this method has a high probability of mistakenly labeling the correct choice as a negative sample. Similarly, PRD [45] uses contrastive learning to pull the first two rows of an RAVEN problem closer, while pushing them apart from context images sampled from other RAVEN problems. This approach also fails when other RAVEN problems share the same logical rules as the current one.

To avoid the inclusion of correct rules as negative examples, we exclude the last row of RAVEN problems. Our positive samples $\boldsymbol{X}^{pos}$ consist only of context features from the current RAVEN problem, ensuring that these features contain complete and correct rules (as shown in **Fact (1)**). Negative samples are constructed by replacing the features of context image 6 with the features of a choice image. This replacement disrupts the rule embedded in the first two rows, ensuring that the negative sample is valid. To maintain consistency with the original RAVEN test, we replace context image 6 because observers predict the last image by learning from all previous images. Notably, this method to construct positive and negative samples can also be readily used during inference. In summary, our utilization of PEs provides an unified framework for learning and inference.

C. Analysis of PredRNet and SSPredRNet

Our models are centered around PEs in the hidden space. One intriguing question is whether an interpretable interface can be built to visualize the model's predictions. In other words, it would be beneficial to reconstruct the visual appearance of both the predictions and PEs. However, we recognize that directly reconstructing predicted features is challenging, as it heavily depends on the decoder's design. Therefore, we focus on visualizing the hidden representations of object attributes and relationships in both PredRNet and SSPredRNet, as shown in Fig. A1 in Appendix, available online.

Moreover, we also used GradCAM [89] to visualize the representation of relations and performed the decoding analysis. The results support that our model indeed learned the representations of abstract rules (Details in Appendix B, available online).

D. Predictive Coding and Neuroscience

Predictive coding is a well-established theory in neuroscience and has been used to explain various phenomena, such as contextual effects in visual processing [18] and dopaminergic activity in reinforcement learning [51]. However, no research has yet directly applied predictive coding to explain high-level visual reasoning in humans. In Rao's seminal work [18], predictive coding was implemented using neural networks. In the era of deep learning, more studies have integrated predictive coding theories, for instance, by adding feedback connections in feedforward CNNs [75], [76] and RNNs [63] for object recognition. Research has also shown that predictive coding can lead to neural activity in artificial networks that closely resembles that of biological neurons [77], [90].

Here, we include no additional feedback or recurrent connections. Instead, we use PE as a key feature to deduce abstract

rules or to construct positive/negative samples. Interestingly, although we do not explicitly minimize PEs, the PE in PredRNet naturally decreases during optimization. This emergent neural phenomenon aligns with findings in recent neuroAI studies (e.g., grid cells, V1 simple cells). To our best knowledge, this is the first example where PE minimization has naturally emerged during model training. These results highlight deep connections between artificial and biological neural networks.

E. Predictive Coding and Intelligence

AVR is considered as a crucial step toward general intelligence. But does predictive coding represent a fundamental computational mechanism for realizing general intelligence? Understanding the nature of intelligence is a shared goal for both neuroscientists and AI researchers, though it remains a broad and complex challenge.

We believe predictive coding is one specific implementation of the broader Bayesian brain theory, but it is certainly not the only approach to performing inference. For example, Karl Friston recently proposed that minimizing free energy could serve as a fundamental computational principle for all living organisms [60]. In this context, free energy represents the gap between an organism's interpretation of the external world and observed events. As this theory evolves, concepts like active inference have been introduced to explain human behavior. Therefore, we view predictive coding as highly consistent with the more comprehensive Bayesian brain theory. We believe that the combination of Bayesian inference and neural networks may be the foundational framework for realizing intelligence.

While our framework is developed for AVR, its central principle—minimizing prediction errors (PEs)—closely parallels the learning dynamics of modern large language models (LLMs). In autoregressive and masked prediction tasks, LLMs are trained to reduce the gap between predicted and actual tokens, effectively minimizing an internal error signal derived from contextual expectations. This process resonates with predictive coding theories in neuroscience. There is neural evidence that such prediction-based learning also parallels neural processing in the brain. In our case, stacked predictive reasoning blocks apply the same principle of PE minimization to iteratively resolve abstract visual rules. Although the domains and architectures differ, this shared reliance on error-driven learning suggests a deeper unifying mechanism across cognitive neuroscience, visual reasoning, and large-scale AI. We believe these parallels merit further exploration and may open new avenues at the intersection of biologically inspired computation and next-generation AI systems.

F. Limitations

Despite strong empirical results and grounding in neuroscience, our PE-based framework has several limitations. While the models generalize well to structured reasoning datasets and show encouraging results on Bongard-HOI, handling the full complexity of natural scenes remains an open challenge. The unsupervised variant provides a biologically inspired alternative to label-based learning, but its performance still falls short of the supervised counterpart, suggesting the need for stronger inductive biases or staged learning strategies. In addition, the fixed hierarchical prediction–matching design improves interpretability but may limit flexibility for tasks requiring more adaptive reasoning. Future work could explore dynamic architectures or attention-driven error routing. Finally, although our approach qualitatively resembles dopamine-based error signaling, its biological plausibility is preliminary, and incorporating richer neural constraints (e.g., temporal dynamics, neuromodulation) would strengthen this connection. These limitations point to key opportunities for building more general, scalable, and biologically grounded reasoning systems.

REFERENCES

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

[2] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1646–1654.

[3] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 06, pp. 1137–1149, Jun. 2017.

[6] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[7] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[10] C. Zhang, F. Gao, B. Jia, Y. Zhu, and S.-C. Zhu, "Raven: A dataset for relational and analogical visual reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5317–5327.

[11] D. Barrett, F. Hill, A. Santoro, A. Morcos, and T. Lillicrap, "Measuring abstract reasoning in neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 511–520.

[12] S. S. Mondal, T. W. Webb, and J. Cohen, "Learning to reason over visual objects," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 35764–35785.

[13] F. Hill, A. Santoro, D. Barrett, A. Morcos, and T. Lillicrap, "Learning to make analogies by contrasting abstract relational structure," in *Proc. Int. Conf. Learn. Representations*, 2019.

[14] W. Nie, Z. Yu, L. Mao, A. B. Patel, Y. Zhu, and A. Anandkumar, "Bongard-logo: A new benchmark for human-level concept learning and reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 16468–16 480.

[15] H. Jiang, X. Ma, W. Nie, Z. Yu, Y. Zhu, and A. Anandkumar, "Bongard-HOI: Benchmarking few-shot visual reasoning for human-object interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 19056–19065.

[16] J. C. Raven and J. Court, *Raven's Progressive Matrices*. Los Angeles, CA, USA: Western Psychological Services, 1938.

[17] M. M. Bongard, *Pattern Recognit.*, Rochelle Park, N.J.: Hayden Book Co., Spartan Books. (Original publication: Проблема Узнавания, Nauka Press, Moscow, 1967), 1970.

[18] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects," *Nature Neurosci.*, vol. 2, no. 1, pp. 79–87, 1999.

[19] M. W. Spratling, "Predictive coding as a model of cognition," *Cogn. Process.*, vol. 17, no. 3, pp. 279–305, 2016.

[20] L. Yang et al., "Neural prediction errors enable analogical visual reasoning in human standard intelligence tests," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 39572–39 583.

[21] M. Małkiński and J. Mańdziuk, "A review of emerging research directions in abstract visual reasoning," *Inf. Fusion*, vol. 91, pp. 713–736, 2022.

[22] P. A. Carpenter, M. A. Just, and P. Shell, "test What one intelligence test measures: A theoretical account of the processing in the Raven progressive matrices test," *Psychol. Rev.*, vol. 97, no. 3, 1990, Art. no. 404.

[23] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cogn. Sci.*, vol. 7, no. 2, pp. 155–170, 1983.

[24] M. Mitchell, *Analogy-Making as Perception: A Computer Model*. Cambridge, MA, USA: Mit Press, 1993.

[25] K. Wang and Z. Su, "Automatic generation of Raven's progressive matrices," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 903–909.

[26] D. Hoshen and M. Werman, "IQ of neural networks," 2017, *arXiv: 1710.01692*.

[27] A. Santoro et al., "A simple neural network module for relational reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4974–4983.

[28] Y. Benny, N. Pekar, and L. Wolf, "Scale-localized abstract reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12557–12 565.

[29] S. Hu, Y. Ma, X. Liu, Y. Wei, and S. Bai, "Stratified rule-aware network for abstract visual reasoning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1567–1574.

[30] S. Spratley, K. A. Ehinger, and T. Miller, "Unicode analogies: An anti-objectivist visual reasoning challenge," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19082–19091.

[31] C. Zhang, B. Jia, F. Gao, Y. Zhu, H. Lu, and S.-C. Zhu, "Learning perceptual inference by contrasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1075–1087.

[32] T. Zhuo and M. Kankanhalli, "Effective abstract reasoning with dual-contrast network," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 16514–16526.

[33] K. Zheng, Z.-J. Zha, and W. Wei, "Abstract reasoning with distracting features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5842–5853.

[34] W. Zhang et al., "Learning robust rule representations for abstract reasoning via internal inferences," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 33550–33562.

[35] M. Jahrens and T. Martinetz, "Solving Raven's progressive matrices with multi-layer relation networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–6.

[36] S. Spratley, K. Ehinger, and T. Miller, "A closer look at generalisation in Raven," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 601–616.

[37] S. S. Mondal, J. D. Cohen, and T. W. Webb, "Slot abstractors: Toward scalable abstract visual reasoning," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 36088–36105.

[38] C. Zhang, B. Jia, S.-C. Zhu, and Y. Zhu, "Abstract spatial-temporal reasoning via probabilistic auction and execution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9736–9746.

[39] C. Zhang, S. Xie, B. Jia, Y. N. Wu, S.-C. Zhu, and Y. Zhu, "Learning algebraic representation for systematic generalization in abstract reasoning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 692–709.

[40] M. Hersche, M. Zeqiri, L. Benini, A. Sebastian, and A. Rahimi, "A neuro-vector-symbolic architecture for solving Raven's progressive matrices," *Nature Mach. Intell.*, vol. 5, no. 4, pp. 363–375, 2023.

[41] J. Xu, T. Vaidya, Y. Wu, S. Chandra, Z. Lai, and K. F. E. Chong, "Abstract visual reasoning: An algebraic approach for solving Raven's progressive matrices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 6715–6724.

[42] D. Wang, M. Jamnik, and P. Lio, "Abstract diagrammatic reasoning with multiplex graph networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 3260–3279.

[43] M. Małkiński and J. Mańdziuk, "Multi-label contrastive learning for abstract visual reasoning," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[44] Y. Xu et al., "Rulematch: Matching abstract rules for semi-supervised learning of human standard intelligence tests," in *Proc. Int. Joint Conf. Artif. Intell.*, 2023.

[45] N. Q. W. Kiat, D. Wang, and M. Jamnik, "Pairwise relations discriminator for unsupervised Raven's progressive matrices," 2020, *arXiv: 2011.01306*.

[46] Q. H. Tao Zhuo and M. Kankanhalli, "Unsupervised abstract reasoning for Raven's problem matrices," *IEEE Trans. Image Process.*, vol. 30, pp. 8332–8341, 2021.

[47] D. J. Chalmers, R. M. French, and D. R. Hofstadter, "High-level perception, representation, and analogy: A critique of artificial intelligence methodology," *J. Exp. Theor. Artif. Intell.*, vol. 4, no. 3, pp. 185–211, 1992.

[48] K. Saito and R. Nakano, *A Concept Learning Algorithm With Adaptive Search*. New York, NY, USA: Oxford Univ. Press, 1996, pp. 347–363.

[49] S. Depeweg, C. A. Rothkopf, and F. Jäkel, "Solving bongard problems with a visual language and pragmatic reasoning," 2018, *arXiv: 1804.04452*.

[50] S. Kharagorgiev, *Solving Bongard Problems with Deep Learn.*, 2018. [Online]. Available: https://k10v.github.io/2018/02/25/Solving-Bongard-problems-with-deep-learning

[51] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, 1997.

[52] M.V. Srinivasan, S. B. Laughlin, and A. Dubs, "Predictive coding: A fresh view of inhibition in the retina," in *Proc. Roy. Soc. London. Ser. B. Biol. Sci.*, vol. 216, no. 1205, pp. 427–459, 1982.

[53] Y. Dan, J. J. Atick, and R. C. Reid, "Efficient coding of natural scenes in the lateral geniculate nucleus: Experimental test of a computational theory," *J. Neurosci.*, vol. 16, no. 10, pp. 3351–3362, 1996.

[54] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, 1968.

[55] J. Bolz and C. D. Gilbert, "Generation of end-inhibition in the visual cortex via interlaminar connections," *Nature*, vol. 320, no. 6060, pp. 362–365, 1986.

[56] R. Desimone and S. J. Schein, "Visual properties of neurons in area v4 of the macaque: Sensitivity to stimulus form," *J. Neuriophysiol.*, vol. 57, no. 3, pp. 835–868, 1987.

[57] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.

[58] M. R. Mehta, "Neuronal dynamics of predictive coding," *Neuroscientist*, vol. 7, no. 6, pp. 490–495, 2001.

[59] C. Summerfield, T. Egner, M. Greene, E. Koechlin, J. Mangels, and J. Hirsch, "Predictive codes for forthcoming perception in the frontal cortex," *Science*, vol. 314, no. 5803, pp. 1311–1314, 2006.

[60] K. Friston and S. Kiebel, "Predictive coding under the free-energy principle," *Philos. Trans. Roy. Soc. B: Biol. Sci.*, vol. 364, no. 1521, pp. 1211–1221, 2009.

[61] J. Hohwy, A. Roepstorff, and K. Friston, "Predictive coding explains binocular rivalry: An epistemological review," *Cognition*, vol. 108, no. 3, pp. 687–701, 2008.

[62] W. H. Alexander and J. W. Brown, "Frontal cortex function as derived from hierarchical predictive coding," *Sci. Rep.*, vol. 8, no. 1, pp. 1–11, 2018.

[63] Z. Pang, C. B. O'May, B. Choksi, and R. VanRullen, "Predictive coding feedback results in perceived illusory contours in a recurrent neural network," *Neural Netw.*, vol. 144, pp. 164–175, 2021.

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

[65] J. Schmidhuber and S. Heil, "Sequential neural text compression," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 142–146, Jan. 1996.

[66] B. S. Atal and M. R. Schroeder, "Adaptive predictive coding of speech signals," *Bell Syst. Tech. J.*, vol. 49, no. 8, pp. 1973–1986, 1970.

[67] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[68] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

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

[70] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *International Society for Music Information Retrieval*. San Diego, CA, USA: Citeseer, 2011.

[71] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1058–1067.

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

[73] Z. Lin, J. Sun, J.-F. Hu, Q. Yu, J.-H. Lai, and W.-S. Zheng, "Predictive feature learning for future segmentation prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7365–7374.

[74] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[75] B. Choksi, M. Mozafari, C. Biggs O'May, B. Ador, A. Alamia, and R. VanRullen, "Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 14069–14083.

[76] H. Wen, K. Han, J. Shi, Y. Zhang, E. Culurciello, and Z. Liu, "Deep predictive coding network for object recognition," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5266–5275.
[77] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1956–1973.
[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[80] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[81] R. Zhang, "Making convolutional networks shift-invariant again," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7324–7334.
[82] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 539–546.
[83] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1655–1668, Jul. 2019.
[84] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2901–2910.
[85] Y. Wu, H. Dong, R. Grosse, and J. Ba, "The scattering compositional learner: Discovering objects, attributes, relationships in analogical reasoning," 2020, *arXiv: 2007.04212*.
[86] K. Zhao, C. Xu, and B. Si, "Learning visual abstract reasoning through dual-stream networks," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 16979–16988.
[87] S. Shekhar and G. W. Taylor, "Neural structure mapping for learning abstract visual analogies," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop*, 2022.
[88] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.
[89] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[90] W. Lotter, G. Kreiman, and D. Cox, "A neural network trained for prediction mimics diverse features of biological neurons and perception," *Nature Mach. Intell.*, vol. 2, no. 4, pp. 210–219, 2020.

Lingxiao Yang (Member, IEEE) received the PhD degree from the Hong Kong Polytechnic University, China, in 2020. He is currectly an associate professor with Sun Yat-sen University. His research interests include computer vision and machine learning with a focus on object recognition, video understanding, and brain-inspired computational model. He has authored or coauthored about 40 papers in prestigious international journals and conferences.

Xiaohua Xie (Member, IEEE) received the BS degree in mathematics and applied mathematics from Shantou University, in 2005, and the MS degree in information and computing science and the PhD degree in applied mathematics from Sun Yat-sen University, China, in 2007 and 2010, respectively. He was an associate professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. He is currently a professor with Sun Yat-sen University. He has authored or coauthored more than 70 papers in prestigious international journals and conferences. His current research interests include image processing, computer vision, pattern recognition, and computer graphics.

Wei-Shi Zheng is now a full professor with Sun Yat-sen University. His research interests include person/object association and activity understanding in visual surveillance, and the related large-scale machine learning algorithm. He has ever served as area chairs of ICCV, CVPR, ECCV, BMVC, IJCAI and etc. He is associate editors/on the editorial board of *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Artificial Intelligence Journal*, *Pattern Recognition*. He has ever joined Microsoft Research Asia Young Faculty Visiting Programme. He is a Cheung Kong Scholar Distinguished Professor, a recipient of the Excellent Young Scientists Fund of the National Natural Science Foundation of China, and a recipient of the Royal Society-Newton Advanced Fellowship of the United Kingdom.

Fang Fang received the PhD degree in cognitive and biological psychology from the University of Minnesota, in 2006. He is professor of psychology and executive associate director of the IDG/McGovern Institute for Brain Research at Peking University. He was a postdoctoral research associate between 2006 and 2007. His research seeks to understand the neural mechanisms of visual and cognitive processes by combining neuroimaging, electrophysiology, brain stimulation, psychophysics, computational modeling, and human genetics. Topics under investigation include visual learning and adaptation, visual attention and awareness, object and face perception. He received the Young Investigaor Award: Basic Science from the International Union of Psychological Science (IUPsyS) in 2016 and was elected as a fellow of the Association for Psychological Science (APS) in 2018. He currently serves on the editorial board for Current Biology, Experimental Brain Research, and Science China: Life Sciences.

Ru-Yuan Zhang received the BS degree in psychology from Peking University, in 2010, and the PhD degree in brain and cognitive sciences from the University of Rochester, USA, in 2016. He is currently an associate professor with Brain Health Institute, National Center for Mental Disorders, Shanghai Mental Health Center, Shanghai Jiao Tong University School of Medicine and School of Psychology. He has authored or co-authored more than 40 papers in prestigious international journals and conferences in both neuroscience and machine learning, including Nature Human Behavior, PNAS, and ICML. His current interests include human visual neuroscience, human decision making and learning, computer vision, and machine reasoning.

Appendix for Neural Prediction Errors as A Unified Cue for Abstract Visual Reasoning

Lingxiao Yang, Xiaohua Xie, Wei-Shi Zheng, Fang Fang*, Ru-Yuan Zhang*

A. Different Image Encoders

TABLE I: Ablation studies with different image encoders on three RAVEN datasets, including the original RAVEN (RVN-O), RAVEN-FAIR (RVN-F), and Impartial-RAVEN (I-RVN) datasets. ResNet-XE and ResNet-XL different fusion methods, with E for early fusion at the first convolutional layer and L for late fusion in the output of image encoders.

	ResNet-18E	ResNet-50E	ResNet-18L	ResNet-50L	**ResNet-4B**
RVN-O	58.0	61.8	53.7	68.1	57.3
RVN-F	15.8	17.3	77.6	62.7	75.6
I-RVN	15.8	11.8	54.2	68.7	71.5
Avg	29.9	30.3	61.8	66.5	68.1

As in our presentation in the main text, 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 layer (*i.e.*, early fusion). We argue that the images should instead be processed in parallel and transformed into high-level feature embeddings first, 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. We also include our image encoder — ResNet-4B. Although the image encoder is not our main contribution here, we would like to provide some empirical results that might be helpful to this area.

All comparison results are shown in **Table I**. The early 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 performance improvement. Our ResNet-4B contains only 1.28 M parameters (*vs*. 23.8 M in ResNet-50L), but achieves the best overall performance. In addition, according to Table I, ResNet-4B, even as a baseline, outperforms many existing models by a wide margin.

B. Analysis of Our PredRNet and SSPredRNet

Although PredRNet and SSPredRNet achieve impressive performance on multiple benchmarks, there still remains unclear whether PEs encode rule-related information, and whether PEs in supervised PredRNet and unsupervised SSPredRNet encode similar things. We therefore analyze the low-dimensional embedding of PEs in the final PRB of PredRNet and SSPredRNet using t-SNE [1].

In Fig. 1, we present t-SNE visualizations of PEs, with respect to object attributes and relations, since object attributes and relations play important roles in learning abstract rules for AVR. Comparing Figs. 1a and 1b, we observe that while PredRNet generates a greater number of smaller clusters, the boundaries between these clusters remain relatively distinct. In contrast, SSPredRNet exhibits a higher degree of mixing among clusters. This observation helps explain why the performance of unsupervised SSPredRNet is significantly lower than that of supervised PredRNet. Furthermore, comparing the two images in Fig. 1a, we find that PredRNet more effectively encodes the attributes of the objects, with clearer boundaries between attribute clusters. In Fig. 1b, we note that the PEs in SSPredRNet tend to express relations; for example, in the right panel of Fig. 1b, the clusters for "XOR" and "OR" are distinctly visible. This may be related to the directly optimization of PEs, leading SSPredRNet to relatively better learn relations.

Moreover, we analyze SSPredRNet on the PGM-Interpolation dataset using GradCAM [2]. GradCAM allows for generating different activation maps for the same image under different labels. By using the rule labels in the dataset as guidance, we obtained rule-specific activation patterns. As shown in Fig. 2, distinct rules yield different attention behaviors: for Line "AND", the model primarily attends to background line structures (e.g., circles and slanted lines), while for Location "XOR", the model shifts its focus to the top-right circle, which is critical under the XOR relation. Nonetheless, given the complexity of the rules (object category, shape, edge existence, edge number, etc.), activation maps alone cannot fully capture whether the model precisely attends to the rule-defining elements. To provide

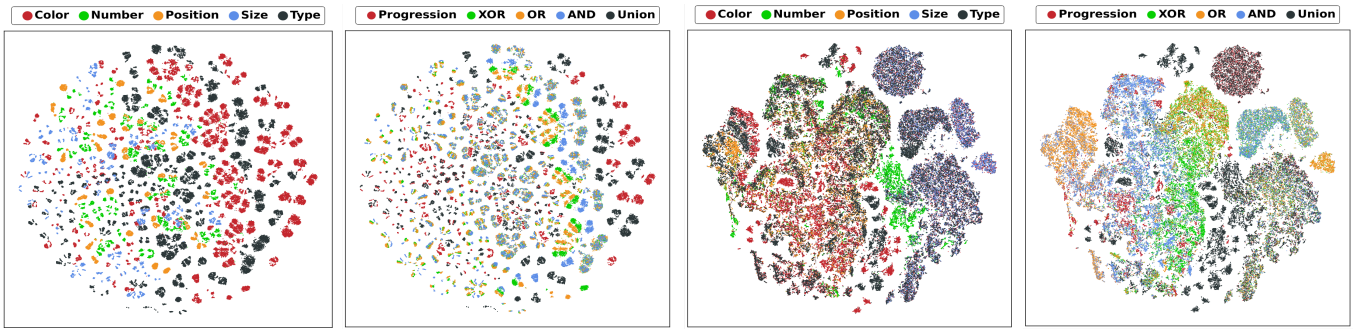
Lingxiao Yang is with School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou, China (e-mail: yanglx9@mail.sysu.edu.cn)

Xiaohua Xie and Wei-Shi Zheng are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China, also with the Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, MOE, Guangzhou, China

Fang Fang is with School of Psychological and Cognitive Sciences and Beijing Key Laboratory of Behavior and Mental Health, IDG/McGovern Institute for Brain Research, Peking-Tsinghua Center for Life Sciences, and Key Laboratory of Machine Perception, MOE, Peking University, 100871, Beijing, China.

Ru-Yuan Zhang is with Brain Health Institute, National Center for Mental Disorders, Shanghai Mental Health Center, Shanghai Jiao Tong University School of Medicine and School of Psychology, Shanghai, China (e-mail: ruyuanzhang@sjtu.edu.cn).

∗ corresponding authors.



(a) PredRNet: **Left:** object attributes, **Right:** relations.

(b) SSPredRNet: **Left:** object attributes, **Right:** relations.

Fig. 1: T-SNE of PEs of the 3-rd PRB. Colored with respect to different object attributes (*Color*, *Number*, *Position*, *Size* and *Type*) and relations (*Progression*, *XOR*, *OR*, *AND* and *Union*) in PGM-Neutral.

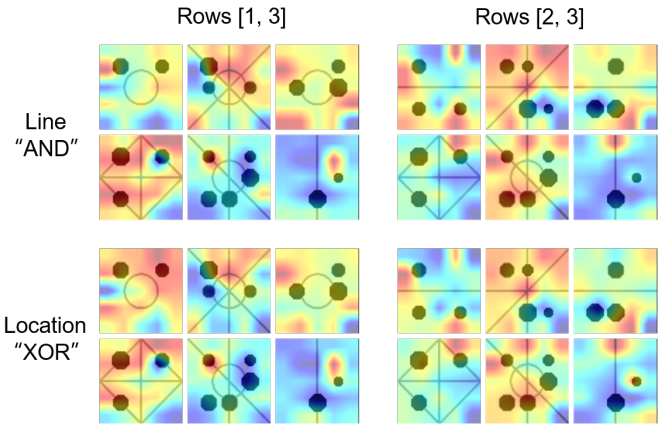


Fig. 2: GradCAM on an problem from PGM-Interpolation. "Rows [X,3]" means that using the X-th and third rows for reasoning, which is consistent with our SSPredRNet.

a more quantitative assessment, we further froze the trained SSPredRNet and trained a linear classifier on its outputs using the 12 rule categories provided in the dataset's meta information. On the PGM-Interpolation split (train/test), the classifier achieved 55.4% accuracy, substantially higher than the random baseline of 8.3%. These analyses together suggest that SSPredRNet does encode rule-relevant information, supporting its ability to capture compositional abstractions.

C. Explicit Rule Modeling in PRBs

TABLE II: Comparing our proposed PRB with a variant by modeling rules explicitly in PRB.

	RAVEN-O	RAVEN-F	I-RAVEN	Avg
Explicit Rule	92.8	96.8	94.4	94.7
Our PRB	96.7	97.9	97.2	97.3

In our current method, we do not explicitly learn rules, but the PRB does involve an implicitly rule-based representation process. Taking the RAVEN problem as an example, our PRB's first uses an 8×1 convolution layer to directly combine information from Panels $1-8$ and matches it with Panels $9-16$. Clearly, this combination process includes both accurate rules $[1, 2, 3]$ and $[4, 5, 6]$, as well as uncertain information, such as $[2, 3, 4]$. We do not explicitly specify the range of combination. To enable comparison, we artificially modify the combination range by replacing the prediction convolution in the PRB with a $3x1$ convolutional layer, aggregating only $[1, 2, 3]$ and $[4, 5, 6]$, and then averaging the information from both rows for the prediction of the third row. The other parts of the model remain unchanged. Results are shown in Table II. We did not find that exclusively include accurate rules is superior to our current PRB approach.

D. Explicit Minimization of PEs

TABLE III: Explicit minimization of PEs in our PredRNet.

	RAVEN-O	RAVEN-F	I-RAVEN	Avg
PredRNet	96.7	97.9	97.2	97.3
PredRNet + PE loss	85.7	91.1	89.3	88.7

We additionally incorporate a loss to minimize errors. However, we observed a slight decrease in accuracy, as shown in Table III. This may because SSPredRNet is less accurate than PredRNet in representing object attributes and the relationships between objects, as the T-SNE shown in Fig. 1. Adding this less accurate representation to the already precise outputs can indeed disrupt the original expression. Furthermore, the introduction of multiple loss functions raises the question of how to effectively balance these losses.

E. Compositional Generalization Evaluation on RAVEN-FAIR

Since our focus is on rule understanding and reasoning, we performed the dataset split primarily along the rule dimension. Specifically, RAVEN-FAIR [3], [4] contains multiple attributes (e.g., object size and object category such as triangles or circles) governed by four major rules: Constant, Progression, Arithmetic, and Distribute Three.

To evaluate compositional generalization, we designed the split such that the training set includes only two rules for each attribute — Constant and Progression for object size, and

Arithmetic and Distribute Three for object category — while the test set includes the complementary rules not seen during training, *i.e.*, Arithmetic and Distribute Three for object size, and Constant and Progression for object category.

Importantly, this split does not constrain the data to single-rule instances. Within these restrictions, different attributes and rules can still combine freely. For instance, in the training set, object size governed by (Constant, Progression) may co-occur with object category governed by (Arithmetic, Distribute Three), whereas in the test set, the combinations are reversed. Thus, both the training and test sets contain diverse cross-attribute compositions, while ensuring that specific combinations do not overlap between them.

Under this split, SSPredRNet achieves an accuracy of 60.3%, slightly lower than the 66.5% achieved on the original RAVEN-FAIR setting, further demonstrating its robustness to compositional shifts.

REFERENCES

[1] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[2] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 618–626.

[3] Y. Benny, N. Pekar, and L. Wolf, "Scale-localized abstract reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 557–12 565.

[4] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5317–5327.