Supplementary Materials: Hierarchical Perceptual and Predictive Analogy-Inference Network for Abstract Visual Reasoning

Anonymous Authors

1 DETAILED DATASET DESCRIPTIONS

The proposed method is systematically evaluated over four benchmark datasets [1, 2, 4, 8]. The examples of the four datasets are shown in Fig. 2 and different configurations for RAVEN-style datasets [2, 4, 8] are illustrated in Fig. 1.

1.1 PGM dataset.

Barrett *et al.* [1] developed the first large-scale RPM benchmark PGM (Procedurally Generated Matrices) evaluating the analogical visual reasoning ability, which contains 8 different regimes and each having 1.42M question samples with 22.72M images. It has diverse analogical rules (*i.e.*, `AND`, `OR`, `XOR`, `Union` and `Progression`) among different attributes of shape and/or line objects, including `Size`, `Color`, `Number`, `Position` and `Type`. Most existing methods are evaluated on the Neutral regime, where the training and test sets are sampled from the same distribution of Neutral regime. Other regimes are used to evaluate the generalizability where different types of rules are omitted from the training and validation sets and the test set consists solely of the held-out rules.

1.2 Original RAVEN dataset.

Zhang *et al.* [8] developed the benchmark RPM dataset named RAVEN with 70K question sets, where each contains 8 question images and 8 candidate answer images. The candidates are generated by permuting the answer image by randomly shifting one attribute value. The dataset is equally distributed into 7 configurations, *i.e.*, `Center`, `2x2Grid`, `3x3Grid`, `Left-Right`, `Up-Down`, `Out-InCenter`, and `Out-InGrid`. Each question contains 6 visual attributes (`Angle`, `Number`, `Position`, `Type`, `Size` and `Color`) and 4 analogical rules (`Constant`, `Progression`, `Arithmetic` and `Distribute_Three`). Extra noise is added to attributes to further challenge the solvers. Compared to the PGM dataset, the number of rule instances and structural combinations has doubled, and the data is only one twentieth of it, which is more challenging.

1.3 Impartial-RAVEN and RAVEN-FAIR datasets.

Due to the shortcut in the answer generation process of the original RAVEN [8], the aggregation of the most common values for each attribute could be the correct answer. Hu *et al.* [4] and Benny *et al.* [2] fixed the loophole of the original RAVEN and developed two RAVEN variants, *i.e.*, I-RAVEN (Impartial-RAVEN) and RAVEN-FAIR datasets, respectively. In the I-RAVEN dataset, the negative candidate answers are generated by hierarchically permuting one attribute of the ground-truth answer in three iterations. For each iteration, two child nodes are generated, where one node remains the same as the parent node while the other permutes one attribute. The RAVEN-FAIR dataset iteratively enlarges the answer set starting with the correct answer only and changing one attribute value from either the correct answer or a generated negative answer. Except for the answer generation, other settings remain the same as in the original RAVEN for both variants.

1.4 Comparisons of Three RAVEN Datasets.

The differences between three RAVEN datasets [2, 4, 8] are demonstrated through an illustrative example shown in Fig. 2. The three samples share the same question images, while the answer images are generated through different schemes.

- The loophole in the original RAVEN [8] is shown in Fig. 2b that the correct answer can be derived by aggregating the most common attributes in the answer set, *i.e.*, hexagon shape, medium-gray color and medium size.
- As shown in Fig. 2c, the I-RAVEN dataset [4] fixes the loophole by hierarchically permuting one attribute from the correct answer, *i.e.*, the answer set in the I-RAVEN dataset has balanced attributes for the overall eight options (two shape types, two sizes and two colors), and the correct answer can no longer be derived by aggregating the most common attributes in the answer set as in the original RAVEN.
- As shown in Fig. 2d, the RAVEN-FAIR dataset [2] rectifies the loophole by generating the options with more randomness. Compared with the I-RAVEN dataset, it has more attribute diversities, while keeping the principle that the attributes for the correct answers can't be derived by majority voting.

2 STATE-OF-THE-ART METHODS

The proposed method is compared with the following state-of-the-art methods on these four datasets: CoPINet [9], WReN [1], DCNet [10], SRAN [4], MRNet [2], AlgebraicMR [6], HCV-ARR [3], STSN [5] and PredRNet [7].

CoPINet [9] models the probability of each candidate answer by applying a contrasting module.

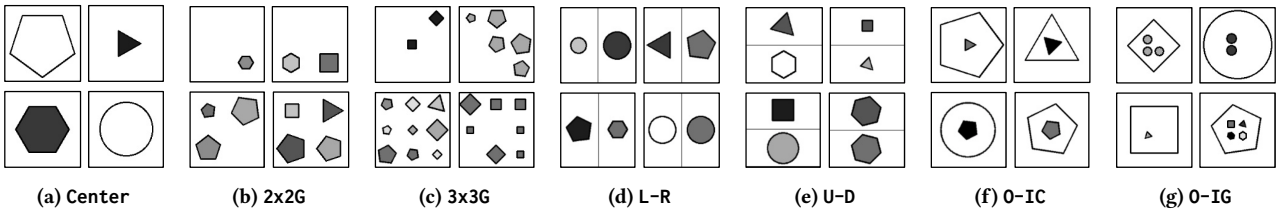
WReN [1] applies a Relation Network to model the pairwise interactions between questions and answers, and utilizes the constructive metadata as the auxiliary loss for reasoning.

DCNet [10] consists of a rule contrast module and a choice contrast module to exploit the inherent structure of RPMs, which compares the latent rules among rows to increase the differences among the options.

SRAN [4] utilizes a hierarchical rule embedding module and a gated embedding fusion module to output the rule embedding given two row sequences.

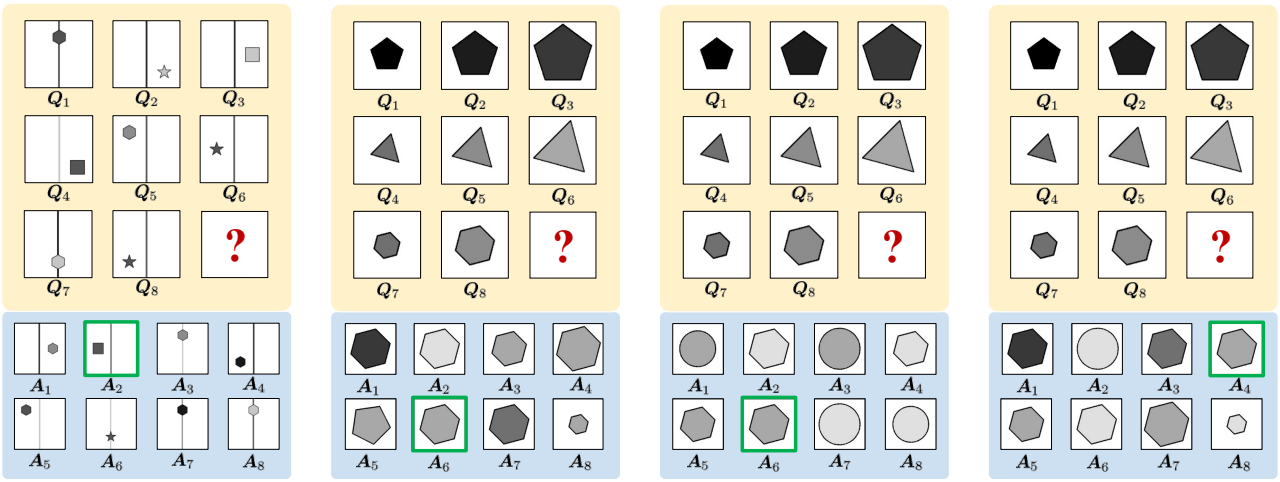
MRNet [2] is established by using multi-resolution convolution layers for visual perception, and computing the row similarity as the inductive reasoner over RPMs.

AlgebraicMR [6] contains an object detector to perceive discrete entity attributes, and utilizes algebraic sub-routines like Gröbner basis and ideal containment to handle RPMs as computational problems.



(a) `Center` (b) `2x2G` (c) `3x3G` (d) `L-R` (e) `U-D` (f) `O-IC` (g) `O-IG`

Figure 1: Examples of seven problem configurations in the RAVEN datasets [2, 4, 8]. Settings `Center`, `Left-Right` (denoted as `L-R`), `Up-Down` (denoted as `U-D`) and `Out-InCenter` (denoted as `O-IC`) have fixed positional layouts of objects, while `2x2Grid` (denoted as `2x2G`), `3x3Grid` (denoted as `3x3G`) and `Out-InGrid` (denoted as `O-IG`) may contain complex relations over the number and position of objects and hence are more challenging to resolve.



(a) PGM dataset [1] (b) Original RAVEN dataset [8] (c) Impartial-RAVEN dataset [4] (d) RAVEN-FAIR dataset [2]

Figure 2: Examples of four benchmark datasets [1, 2, 4, 8]. The correct answers are framed in green. As shown in Fig. 2b-2d, the three RAVEN variants share the same question images, while the answer images are generated through different schemes.

HCV-ARR [3] adapts a mixed model of convolution blocks and vision Transformer blocks to extract multi-level features from RPM images, and then dynamically learns the importance weights over different dimensions of row features based on an attention mechanism.

STSN [5] incorporates the problem-specific inductive biases as an object-centric encoder, and a transformer reasoning module to solve the abstract visual reasoning problems.

PredRNet [7] utilizes consecutive residual convolutional layers to extract high-level visual features from images, and uses convolutional blocks to extract abstract rules to predict the correct answers.

3 DETAILS OF ABLATION STUDIES

The full ablation results are shown in Table 1 and Table 2 for different problem configurations on the three RAVEN datasets [2, 4, 8]. Problem configurations such as `Center`, `L-R`, `U-D` and `O-IC` contain constant object numbers and positions and rules are only applied to object shape types, colors and sizes, while problem configurations such as `2x2G`, `3x3G` and `O-IG` are built based on more complicated relations over all five attributes, including higher-order position relations over objects like rolling-over or binary operations. Hence, the position-related rules in `2x2G`, `3x3G` and `O-IG` are more challenging to handle, even for humans.

The previously best performing method, PredRNet [7], is selected as the baseline, which consists of an image encoder for visual perception and a decoder for analogical reasoning. We substitute the respective modules with the proposed modules and the results are summarized in Table 1. The performance improvements shown in Table 1 are derived by comparing either or both modules with the baseline method. We can see from the average results that applying both HPALC and PredAI individually will result in great performance improvements over all three datasets consistently. When the two modules are jointly utilized, the proposed method HP^{2}AI significantly outperforms PredRNet [7] on all three datasets in terms of average accuracy. The performance gains over `3x3G` and `O-IG` are most significant, illustrating that the proposed HPALC and PredAI are both effective in handling high-level number- and position-related rules.

In Table 2, we analyse the major hyper-parameter settings for the proposed HP^{2}AI, including the number of hierarchical stages J for both HPALC and PredAI modules, and the number of PredAI

Table 1: Ablation studies of the two major components of the proposed HP^{2}AI on three RAVEN datasets [2, 4, 8].

	Major Module		Accuracy (%) on Configurations							
	HPALC	PredAI	**Avg.**	Center	2x2G	3x3G	L-R	U-D	O-IC	O-IG
O-RVN	✗	✗	95.8	99.8	95.1	87.6	99.2	99.4	99.9	89.4
	✓	✗	97.2 (+1.4)	99.8 (+0.0)	97.6 (+2.5)	91.6 (+4.0)	99.6 (+0.4)	99.5 (+0.1)	99.7 (−0.2)	94.0 (+4.6)
	✗	✓	97.7 (+1.9)	100.0 (+0.2)	97.9 (+2.8)	92.9 (+5.3)	99.9 (+0.7)	99.8 (+0.4)	99.8 (−0.1)	93.5 (+4.1)
	✓	✓	98.8 (+3.0)	100.0 (+0.2)	98.8 (+3.7)	95.3 (+7.7)	99.9 (+0.7)	99.8 (+0.4)	99.9 (+0.0)	98.0 (+8.6)
I-RVN	✗	✗	96.5	99.9	97.8	91.2	99.7	99.7	99.6	87.7
	✓	✗	98.3 (+1.8)	99.9 (+0.0)	98.7 (+0.9)	96.9 (+5.7)	99.8 (+0.1)	99.7 (+0.0)	99.6 (+0.0)	95.3 (+7.6)
	✗	✓	98.1 (+1.6)	99.9 (+0.0)	99.3 (+1.5)	94.9 (+3.7)	99.9 (+0.2)	99.9 (+0.2)	100.0 (+0.4)	94.2 (+6.5)
	✓	✓	99.4 (+2.9)	100.0 (+0.1)	99.9 (+2.1)	97.4 (+6.2)	99.9 (+0.2)	100.0 (+0.3)	100.0 (+0.4)	98.8 (+11.1)
RVN-F	✗	✗	97.1	99.8	97.3	92.6	99.7	99.5	99.7	91.2
	✓	✗	98.0 (+0.9)	99.9 (+0.1)	98.8 (+1.5)	95.1 (+2.5)	99.9 (+0.2)	99.9 (+0.4)	99.8 (+0.1)	93.3 (+2.1)
	✗	✓	97.8 (+0.7)	99.9 (+0.1)	98.5 (+1.2)	94.2 (+1.6)	100.0 (+0.3)	99.6 (+0.1)	99.8 (+0.1)	93.1 (+1.9)
	✓	✓	98.6 (+1.5)	100.0 (+0.2)	99.4 (+2.1)	96.9 (+4.3)	99.9 (+0.2)	99.9 (+0.4)	99.7 (+0.0)	94.2 (+3.0)

Table 2: Ablation studies of different hierarchical stages J and different K of PredAI blocks for the proposed HP^{2}AI on all three RAVEN datasets [2, 4, 8].

	Params.	Accuracy (%) on Configurations							
		Avg.	Center	2x2G	3x3G	L-R	U-D	O-IC	O-IG
O-RVN	$J=1$	87.1	98.7	72.5	74.0	99.4	99.2	99.6	66.3
	$J=2$	92.4 (+5.3)	99.5 (+0.8)	77.6 (+5.1)	77.7 (+3.7)	99.9 (+0.5)	99.6 (+0.4)	99.5 (−0.1)	92.8 (+26.5)
	$J=3$	98.8 (+11.7)	100.0 (+1.3)	98.8 (+26.3)	95.3 (+21.3)	99.9 (+0.5)	99.8 (+0.6)	99.9 (+0.3)	98.0 (+31.7)
	$J=4$	98.6 (+11.5)	100.0 (+1.3)	98.8 (+26.3)	93.9 (+19.9)	99.9 (+0.5)	99.8 (+0.6)	99.9 (+0.3)	97.8 (+31.5)
	$K=1$	98.1	99.9	98.1	92.2	99.6	99.6	99.6	97.4
	$K=2$	98.4 (+0.3)	99.9 (+0.0)	98.6 (+0.5)	92.9 (+0.7)	99.8 (+0.2)	99.8 (+0.2)	99.8 (+0.2)	98.0 (+0.6)
	$K=3$	98.8 (+0.7)	100.0 (+0.1)	98.8 (+0.7)	95.3 (+3.1)	99.9 (+0.3)	99.8 (+0.2)	99.9 (+0.3)	98.0 (+0.6)
	$K=4$	98.0 (−0.1)	99.8 (−0.1)	98.3 (+0.2)	91.3 (−0.9)	99.8 (+0.2)	99.7 (+0.1)	99.9 (+0.3)	97.2 (−0.2)
I-RVN	$J=1$	88.9	99.4	72.4	68.1	99.7	99.8	99.8	83.2
	$J=2$	93.6 (+4.1)	100.0 (+0.6)	93.4 (+21.0)	81.5 (+13.4)	99.8 (+0.1)	99.9 (+0.1)	99.6 (−0.2)	76.7 (−6.5)
	$J=3$	99.4 (+10.5)	100.0 (+0.6)	99.9 (+27.5)	97.4 (+29.3)	99.9 (+0.2)	100.0 (+0.2)	100.0 (+0.2)	98.8 (+15.6)
	$J=4$	99.1 (+10.2)	99.9 (+0.5)	99.5 (+27.1)	97.0 (+28.9)	99.9 (+0.2)	99.8 (+0.0)	99.9 (+0.1)	98.6 (+15.4)
	$K=1$	98.6	99.8	98.9	94.1	99.6	99.8	99.7	98.2
	$K=2$	99.1 (+0.5)	99.9 (+0.1)	99.5 (+0.6)	95.4 (+1.3)	99.9 (+0.3)	100.0 (+0.2)	100.0 (+0.3)	99.0 (+0.8)
	$K=3$	99.4 (+0.8)	100.0 (+0.2)	99.9 (+1.0)	97.4 (+3.3)	99.9 (+0.3)	100.0 (+0.2)	100.0 (+0.3)	98.8 (+0.6)
	$K=4$	99.0 (+0.4)	99.9 (+0.1)	99.5 (+0.6)	95.5 (+1.4)	99.8 (+0.2)	99.9 (+0.1)	99.9 (+0.2)	98.0 (−0.2)
RVN-F	$J=1$	91.4	98.8	82.2	81.2	99.7	99.8	99.7	78.8
	$J=2$	96.4 (+5.0)	99.3 (+0.5)	89.7 (+7.5)	89.9 (+8.7)	99.5 (−0.2)	99.9 (+0.1)	99.7 (+0.0)	96.8 (+18.0)
	$J=3$	98.6 (+7.2)	100.0 (+1.2)	99.4 (+17.2)	96.9 (+15.7)	99.9 (+0.2)	99.9 (+0.1)	99.7 (+0.0)	94.2 (+15.4)
	$J=4$	98.5 (+7.1)	99.8 (+1.0)	99.1 (+16.9)	96.5 (+15.3)	99.9 (+0.2)	99.9 (+0.1)	99.9 (+0.2)	94.1 (+15.3)
	$K=1$	98.3	99.8	98.4	95.4	99.7	99.6	99.9	95.5
	$K=2$	98.3 (+0.0)	99.9 (+0.1)	99.1 (+0.7)	96.0 (+0.6)	99.9 (+0.2)	99.7 (+0.1)	99.9 (+0.0)	93.4 (−2.1)
	$K=3$	98.6 (+0.3)	100.0 (+0.2)	99.4 (+1.0)	96.9 (+1.5)	99.9 (+0.2)	99.9 (+0.3)	99.7 (−0.2)	94.2 (−1.3)
	$K=4$	98.2 (−0.1)	99.9 (+0.1)	99.0 (+0.6)	95.8 (+0.4)	99.6 (−0.1)	99.8 (+0.2)	99.8 (−0.1)	93.1 (−2.4)

blocks K in the PredAI module. The performance improvements shown in Table 2 are derived by comparing with $J = 1$ or $K = 1$. On all three datasets, the best values for J and K are 3. Specifically, for the number of hierarchical stages, we can see clear improvements by increasing J from 1 to 3, which can result in over 20% performance gains in settings `2x2G`, `3x3G` and `O-IG`, which corresponds to our design of HPALC that deeper layers are advantageous to capturing high-level spatial semantics. The long-range dependencies in the Patch Attention branch also contribute to such significant performance gains. When the J goes deeper to 4, the reasoning accuracy on all the configurations drops slightly, which is possibly due to the insufficient information in the small feature maps of the deeper stages. Regarding the number of PredAI blocks K, we can see that utilizing even $K = 1$ PredAI block yields competitive results on all the configurations, which demonstrates the effectiveness of the proposed AIB in conducting robust analogical inference by predicting-and-verifying paradigm, and SECA that amplifies the dominant attributes and rules in analogical reasoning. When the PredAI blocks are iterated from $K = 1$ to 3, some initially wrongly induced rules can be rectified during the iteration, which in turn improves the accuracy. When K goes deeper, the model may suffer from over-fitting, which imposes a slight negative impact on the reasoning accuracy.

4 FAILURE CASE ANALYSIS

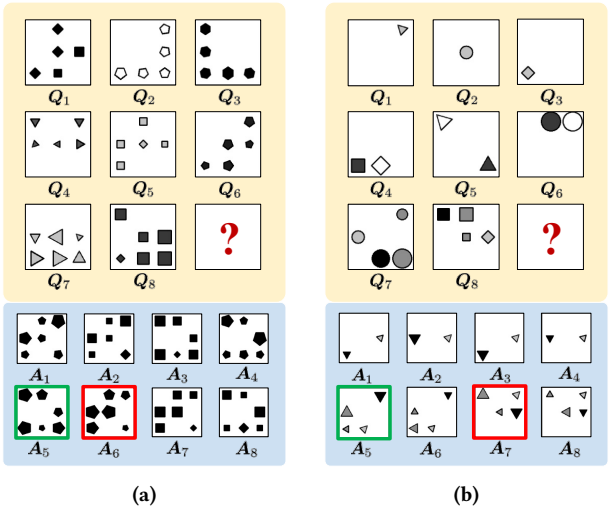


Figure 3: Examples of the failure cases by the proposed HP^{2}AI. The correct answers are framed in green while the answers wrongly predicted by HP^{2}AI are framed in red.

Although the proposed HP^{2}AI achieves near-perfect performances on the four benchmark RPM datasets, it is noticeable that the proposed HP^{2}AI performs slightly poorer on the configurations containing high-level spatial relations, which are also challenging to other methods. We take a step further into these failure cases on the most complicated setting `3x3G` and analyze the underlying reasons. As shown in Fig. 3, the proposed HP^{2}AI fails to correctly predict the answers for those two representative questions, similarly as other methods do. In Fig. 3a, the HP^{2}AI correctly predicts the `Type` attribute and solves the question relying on `Number` attribute. However, Fig. 3a contains an extremely complicated `Progression` rule over the `Position` attribute, which leads to the entities on each panel rolling over the layout. Specifically, the entities in row-wise images perform a rightward cyclic shift ($\rightarrow$), and the boundary entities perform an additional downward cyclic shift ($\rightarrow$+$\downarrow$). The proposed HP^{2}AI fails to model such a complicated relation. Fig. 3b contains another similar `Progression` rule over the `Position` attribute, where the entities in row-wise images perform a leftward cyclic shift together with a downward cyclic shift ($\leftarrow$+$\downarrow$), and the boundary entities perform only the leftward cyclic shift ($\leftarrow$). It is even difficult for humans to induce such a complicated reasoning rule and derive the correct answer for these two questions.

5 MODEL ARCHITECTURE

We provide the detailed architectures and parameters for the proposed HPALC, PredAI and classifier as shown in Tables 3, 4 and 5, respectively. The following table starts with the input of size $(16, 1, 80, 80)$, which represents a complete RPM panel (8 question images + 8 answer images), and each image of size 80×80 pixels.

In Table 3, the local/regional self-attention block (L/R-SA) computes the ***qkv*** multi-head self-attention and has no layer operations, which leads to the same dimensionality for the input and the output. Every output feature map marked in **bold** is passed to the successive block, while simultaneously extracted as multi-level receptive fields and sent into PredAI blocks to conduct relational reasoning.

The architectures and parameters for the proposed Predictive Analogy-Inference block are shown in Table 4. The feature maps from previous HPALC blocks are firstly reshaped, stacked and permuted into a matrix form, and column tensors of first two entities in each row as shown in Eqn. (3) of the manuscript are passed to AIB for prediction and verification.

Lastly, the reasoning features from different stages are jointly passed to the classifier in Table 5 for final prediction, which utilizes the Binary Cross-Entropy as the loss function.

REFERENCES

[1] David G Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. 2018. Measuring Abstract Reasoning in Neural Networks. In *ICML*, Vol. 80. 511–520.

[2] Yaniv Benny, Niv Pekar, and Lior Wolf. 2021. Scale-Localized Abstract Reasoning. In *CVPR*. 12557–12565.

[3] Wentao He, Jialu Zhang, Jianfeng Ren, Ruibin Bai, and Xudong Jiang. 2023. Hierarchical ConViT with Attention-based Relational Reasoner for Visual Analogical Reasoning. In *AAAI*, Vol. 37. 22–30.

[4] Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2021. Stratified Rule-Aware Network for Abstract Visual Reasoning. In *AAAI*, Vol. 35. 1567–1574.

[5] Shanka Subhra Mondal, Taylor Webb, and Jonathan Cohen. 2022. Learning to Reason over Visual Objects. In *ICLR*.

[6] Jingyi Xu, Tushar Vaidya, Yufei Wu, Saket Chandra, Zhangsheng Lai, and Kai Fong Ernest Chong. 2023. Abstract Visual Reasoning: An Algebraic Approach for Solving Raven's Progressive Matrices. In *CVPR*. 6715–6724.

[7] Lingxiao Yang, Hongzhi You, Zonglei Zhen, Dahui Wang, Xiaohong Wan, Xiaohua Xie, and Ru-Yuan Zhang. 2023. Neural Prediction Errors enable Analogical Visual Reasoning in Human Standard Intelligence Tests. In *ICML*, Vol. 202. 39572–39583.

[8] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. RAVEN: A Dataset for Relational and Analogical Visual rEasoNing. In *CVPR*. 5317–5327.

[9] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, HongJing Lu, and Song-Chun Zhu. 2019. Learning Perceptual Inference by Contrasting. In *NeurIPS*, Vol. 32. 1075–1087.

[10] Tao Zhuo and Mohan Kankanhalli. 2021. Effective Abstract Reasoning with Dual-Contrast Network. In *ICLR*.

Table 3: Detailed network architecture for the proposed HPALC, with parameters of channels (C), kernel size (K), stride (S), padding (P) and regional window sizes (W). The sizes of output features at each stage are marked in bold.

Stage J	Module	Layer Operations		Parameters	Input	Output
1	PatchSplit	Conv2d		C64K4S4	(16,1,80,80)	
		LayerNorm		C64		(16,64,20,20)
	PatchAttention Block×2	LayerNorm		C64	(16,64,20,20)	
		L/R-SA		W7		
		Residual				(16,64,20,20)
		LayerNorm		C64	(16,64,20,20)	
		Linear		C64		
		GELU				
		Residual				**(16,64,20,20)**
	LocalContext Block	Conv2d	×3	C32K3P1	(16,1,80,80)	
		BatchNorm		C32		
		ReLU				
		Residual				**(16,64,20,20)**
2	PatchMerge	Unfold		C256	(16,64,20,20)	(16,256,10,10)
		LayerNorm		C256		
		Linear		C128		(16,128,10,10)
	PatchAttention Block×2	LayerNorm		C128	(16,128,10,10)	
		L/R-SA		W7		
		Residual				(16,128,10,10)
		LayerNorm		C128	(16,128,10,10)	
		Linear		C128		
		GELU				
		Residual				**(16,128,10,10)**
	LocalContext Block	Conv2d	×3	C128K3P1	(16,64,20,20)	
		BatchNorm		C128		
		ReLU				
		Residual				**(16,128,10,10)**
3	PatchMerge	Unfold		C512	(16,128,10,10)	(16,512,5,5)
		LayerNorm		C512		
		Linear		C256		(16,256,5,5)
	PatchAttention Block×2	LayerNorm		C256	(16,256,5,5)	
		L/R-SA		W7		
		Residual				(16,256,5,5)
		LayerNorm		C256	(16,256,5,5)	
		Linear		C256		
		GELU				
		Residual				**(16,256,5,5)**
	LocalContext Block	Conv2d	×3	C256K3P1	(16,128,10,10)	
		BatchNorm		C256		
		ReLU				
		Residual				**(16,256,5,5)**

Table 4: Detailed network architecture for the proposed PredAI, with parameters of channels (C), kernel size (K), stride (S) and padding (P). The sizes of output features at each stage are marked in bold.

Stage J	Module		Layer Operations	Parameters	Input	Output
1	DimReduc		Conv2d	C32K1S1	(16,64,20,20)	
			BatchNorm	C32		(16,32,20,20)
	MatReshape		Stack		(16,32,20,20)	(8,9,32,20,20)
			Permute		(8,9,32,20,20)	(8,32,3,3,400)
	AIB	×3	Conv2d	C32K(2,1)S1	(8,32,3,2,400)	
			BatchNorm	C32		
			ReLU			(8,32,3,1,400)
	SECA		AvgPool2d		(8,32,3,1,400)	(8,32,1,1)
			Linear	C2		
			ReLU			
			Linear	C32		(8,32,1,1)
			Scaling		(8,32,3,2,400)	(8,32,3,2,400)
	MLP		Linear	C128	(8,32,3,3,400)	
			Linear	C32		**(8,32,9,400)**
2	DimReduc		Conv2d	C32K1S1	(16,128,10,10)	
			BatchNorm	C32		(16,32,10,10)
	MatReshape		Stack		(16,32,10,10)	(8,9,32,10,10)
			Permute		(8,9,32,10,10)	(8,32,3,3,100)
	AIB	×3	Conv2d	C32K(2,1)S1	(8,32,3,2,100)	
			BatchNorm	C32		
			ReLU			(8,32,3,1,100)
	SECA		AvgPool2d		(8,32,3,1,100)	(8,32,1,1)
			Linear	C2		
			ReLU			
			Linear	C32		(8,32,1,1)
			Scaling		(8,32,3,2,100)	(8,32,3,2,100)
	MLP		Linear	C128	(8,32,3,3,100)	
			Linear	C32		**(8,32,9,100)**
3	DimReduc		Conv2d	C32K1S1	(16,256,5,5)	
			BatchNorm	C32		(16,32,5,5)
	MatReshape		Stack		(16,32,5,5)	(8,9,32,5,5)
			Permute		(8,9,32,5,5)	(8,32,3,3,25)
	AIB	×3	Conv2d	C32K(2,1)S1	(8,32,3,2,25)	
			BatchNorm	C32		
			ReLU			(8,32,3,1,25)
	SECA		AvgPool2d		(8,32,3,1,25)	(8,32,1,1)
			Linear	C2		
			ReLU			
			Linear	C32		(8,32,1,1)
			Scaling		(8,32,3,2,25)	(8,32,3,2,25)
	MLP		Linear	C128	(8,32,3,3,25)	
			Linear	C32		**(8,32,9,25)**

Table 5: Detailed network architecture for the classifier of the proposed HP^{2}AI.

Module	Layer Operations	Parameters	Input	Output
Classifier	AvgPool1d	C1024	(8,32,9,400)	(8,1024)
	AvgPool1d	C1024	(8,32,9,100)	(8,1024)
	AvgPool1d	C1024	(8,32,9,25)	(8,1024)
	Linear	C1024	(8,1024×3)	
	BatchNorm	C1024		
	ReLU			(8,1024)
	Linear	C1	(8,1024)	(8,1)