Human-like Abstract Visual Reasoning via Understanding and Solving Reasoning Loop

Supplementary Material

Table 5. Performance breakdown on the ConceptARC dataset.

Task Category	Accuracy
Top and Bottom 2D	0.53
Extend To Boundary	0.47
Above and Below	0.43
Filled and Not Filled	0.40
Center	0.37
Clean Up	0.37
Complete Shape	0.33
Copy	0.30
Extract Objects	0.30
Horizontal and Vertical	0.27
Inside and Outside	0.27
Move To Boundary	0.23
Top and Bottom 3D	0.20
Order	0.17
Count	0.13
Same and Different	0.13

A. Implementation Details

A.1. Data Augmentation Strategies

We primarily train and evaluate our model on ARC-AGI-1. This dataset consists of 400 training tasks and 400 evaluation tasks, with the grid size of all tasks limited to 30×30. To ensure a fair comparison with TRM [14], we adopt the same dataset configuration and augmentation strategies. Specifically, we use training set and examples of evaluation set to construct the initial training set, resulting in a total of 800 base samples. Subsequently, we apply color permutations, translation transformations, and dihedral group transformations (including random 90-degree rotations, horizontal/vertical flips, and mirror reflections) to each base sample, expanding the sample size by up to 1000 times. To ensure data uniqueness, we deduplicate the repeating samples generated by the different combinations of transformations. Furthermore, consistent with the setup of TRM, we additionally introduce 160 tasks from the ConceptARC dataset and apply the exact same data augmentation pipeline to them.

A.2. Network Architecture Details

The detailed model architecture and feature flow are illustrated in Figure 5. The hidden dimension of the model is set to 368, and the sequence length of the query tokens $\mathbf{Q}$ is 4. UM and SM adopt the Recurrent Transformer structure from TRM [14], with 8 attention heads. The computation processes of UM and SM involve 4 inner loops and 3 outer loops.

Table 6. Accuracy of USRL variants with varying the number of blocks in UM and SM.

UM Blocks	SM Blocks		
	1	**2**	**3**
1	29.6	30.2	30.5
2	29.7	35.3	35.9
3	29.9	37.4	37.8

A.3. Hyperparameters and Training Setup

The hyperparameters and training details for our experiments are as follows: the balancing coefficient λ is set to 0.5, and the global batch size is 768. Due to GPU memory constraints, the training process is optimized using a mini-batch approach. We use the AdamW optimizer with a learning rate of 10^{-4} and a weight decay of 0.01. The model is trained for a total of 100000 epochs. The parameter update phase incorporates the Exponential Moving Average (EMA) method. The overall training hyperparameters and details follow the setup of TRM [14].

B. Additional Experiments

B.1. Analysis on ConceptARC

The ConceptARC [20] dataset covers 16 core concepts of ARC-AGI transformations, aiming to provide a fine-grained evaluation of the model's reasoning capabilities on specific types of ARC-AGI tasks. To rigorously evaluate the performance of USRL, we trained an independent model without including any ConceptARC data and evaluated it on ConceptARC. Through a detailed analysis of the 16 core concept categories, we observe that USRL performs better on geometric and spatial tasks, achieving accuracy in categories such as Top and Bottom 2D (53%), Extend To Boundary (47%), Above and Below (43%), and Filled and Not Filled (40%). However, the model struggles with tasks involving counting and logical comparison, showing lower accuracy in categories like Count (13%), Same and Different (13%), and Order (17%). The specific performance across all task categories is shown in Table 5.

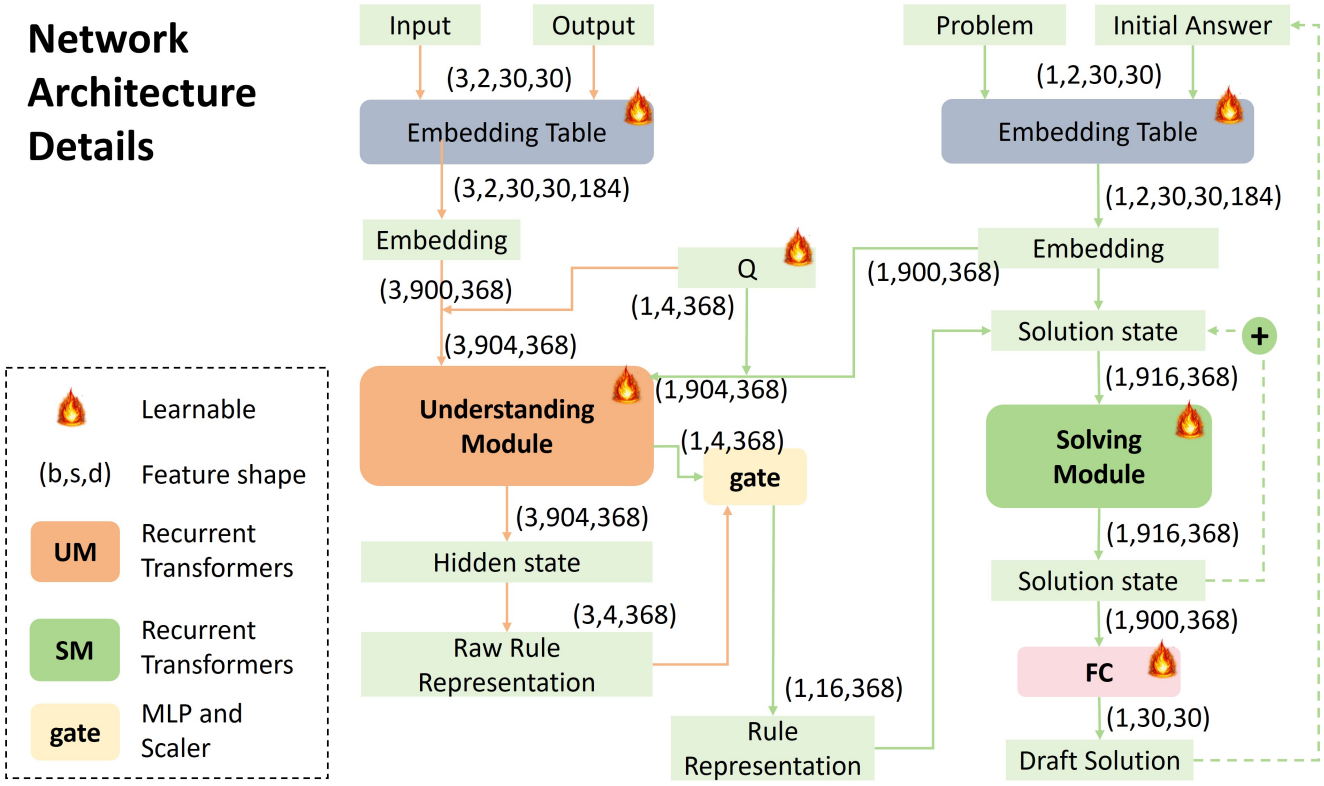

Figure 5. Detailed network architecture and feature flow of the USRL framework.

B.2. Influence of Module Sizes

To analyze how the sizes of UM and SM impact overall performance, we vary the number of transformer blocks in both modules and observe the accuracy. We use a USRL model trained on the ARC-AGI-1 dataset for 20k epochs. Starting from a baseline configuration of 2 UM blocks and 2 SM blocks (with an accuracy of 35.3%), we construct nine model variants with varying block counts and fine-tune each for 10k epochs. As shown in Table 6, when the SM is under-parameterized, the model yields relatively low scores (29.9% for 3UM-1SM and 29.7% for 2UM-1SM). The best performance is achieved with larger, balanced module configurations (37.8% for 3UM-3SM and 37.4% for 3UM-2SM). These results demonstrate that the capacities of UM and SM should be balanced: if SM is under-parameterized, scaling up UM alone is useless. Expanding UM improves performance only when SM possesses sufficient capacity to provide valid feedback.