Using Contrastive Learning for Abstraction and Reasoning task

Gyojoon Gu, Woochang Sim, Jaegyun Im, Sejin Kim, Sundong Kim
Data Science Lab, Gwangju Institute of Science and Technology

1. Introduction

The ARC dataset measures AI's abstract reasoning, contrasting with human performance where humans achieve 80% accuracy to AI's 31%. This gap is largely due to humans' superior prior knowledge. Our research suggests that giving AI information about problem types can bridge this performance gap. To address ARC's data limitations, we employed contrastive learning for effective feature extraction, aiming to enhance ARC problem-solving and contribute to advancements in Artificial General Intelligence (AGI)

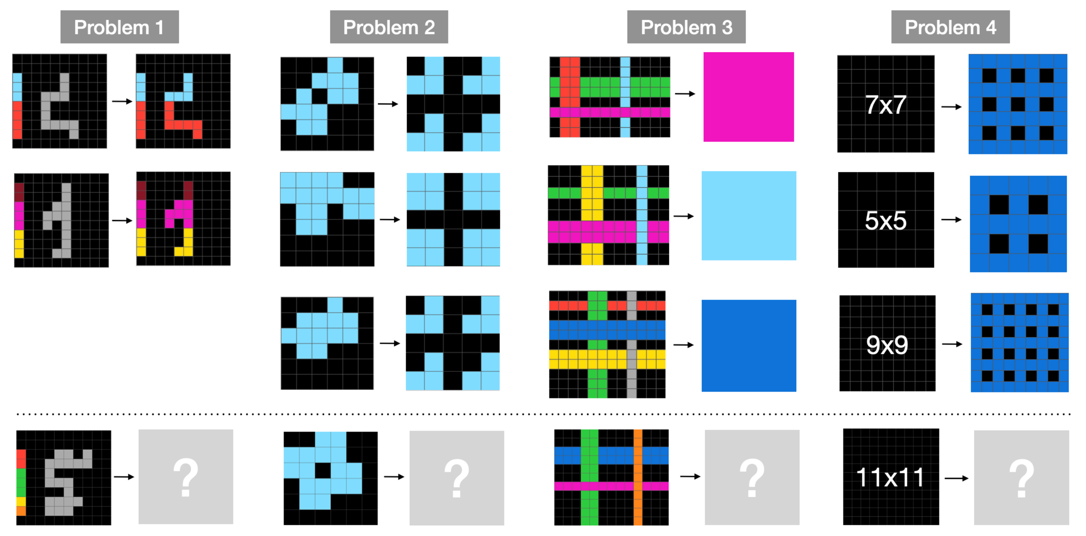

Figure 1: Four different ARC problems.

2. Method

2.1 Contrastive Learning

In our research, contrastive learning is applied to ARC problems using extracted representation vectors, as shown in Figure 2. The approach brings similar problem vectors closer and distances dissimilar ones in the latent space, using cosine similarity. Yellow regions in Figure 3 represent high similarity, while white areas indicate low similarity. Two types of contrastive learning are used: Self-Supervised Learning (SSL) and Supervised Contrastive Learning (SCL), with a cross-entropy loss function.

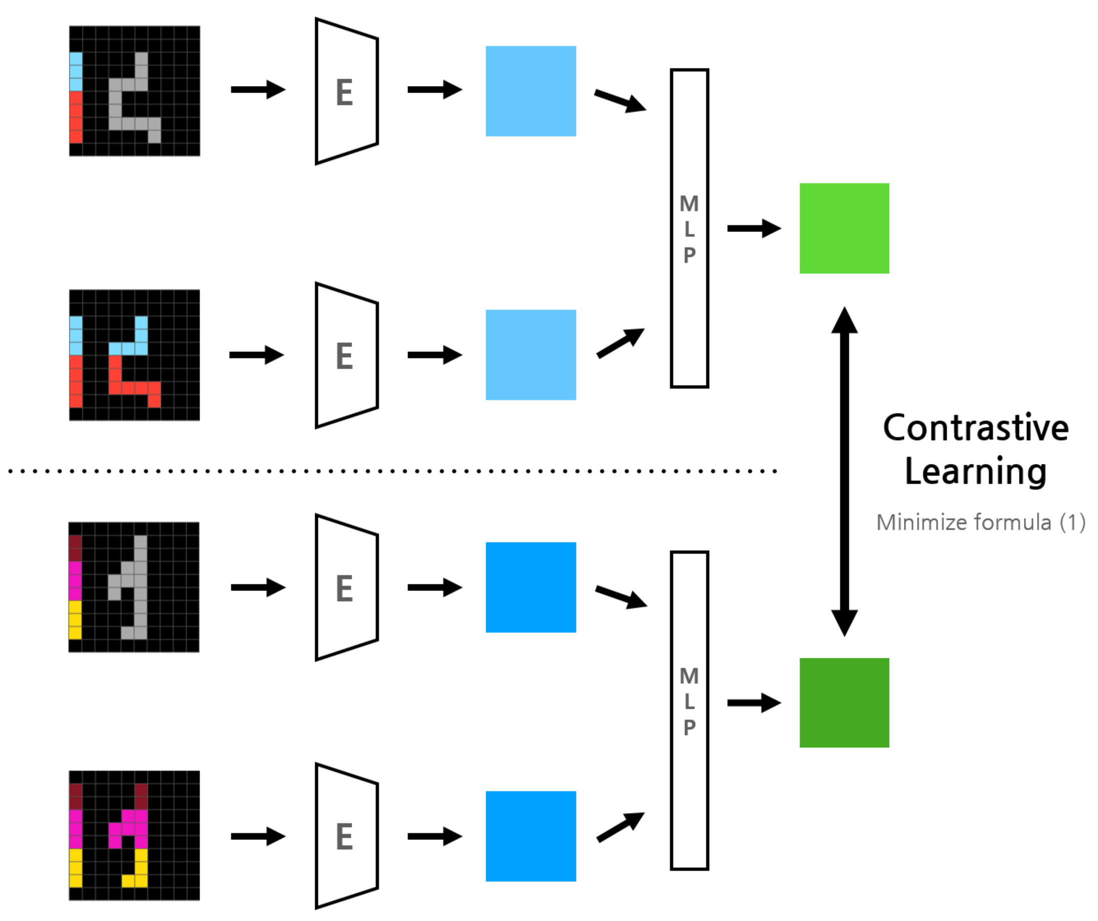

Figure 2: A simplified schematic showing the process of extracting representation vectors for the input-output pairs of ARC problems and conducting contrastive learning.

2.1.1 Self-Supervised Learning

The contrastive learning method in self-supervised learning, as shown in SSL in Figure 3, trains the representation vectors to be distinct from each other during batch training. This method can be used even without labeling for classification classes.

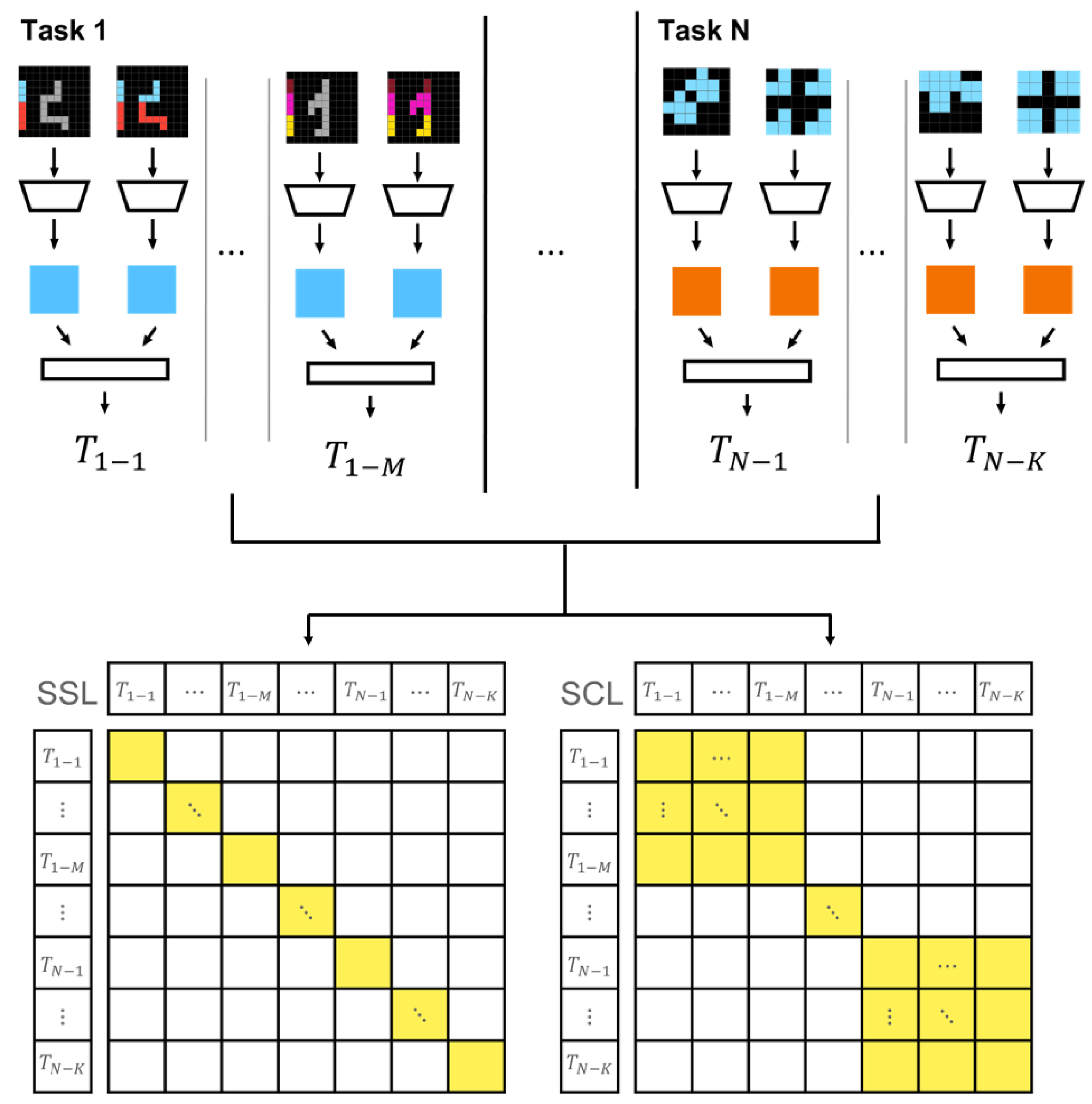

Figure 3: Illustration depicting the process of conducting contrastive learning using representation vectors that represent each problem. In the figure, T_{N-K} represents the representation vector of the Kth input-output pair for the Nth problem.

2.1.2 Supervised Contrastive Learning

Supervised contrastive learning, as in SCL in Figure 3, trains to generate similar representation vectors for examples of the same problem type and differently for other types. Unlike self-supervised learning, this method requires prior labeling for classification classes.

3. Experiment

In our research, classification experiments were conducted using the ARC dataset and the ConceptARC dataset. For these experiments, a VAE encoder was used as the model structure. The loss function for training was the NT-Xent (normalized temperature-scaled cross-entropy loss) used in SimCLR and many previous studies. In addition, KNN and linear probing methods were used to evaluate classification performance before and after applying contrastive learning.

3.1 Dataset

Classification experiments were conducted using the ARC and ConceptARC datasets. The ARC dataset was treated as having 400 distinct problem types. For ConceptARC, 16 predefined classes like 'Above and Below', 'Center', and 'Clean Up' were used for structured experiments.

3.2 Loss Function

When T_{N-K} represents the representation vector for the Kth input-output pair of the Nth problem, and $sim(u,v) = \frac{u \cdot v}{||u||||v||}$ (cosine similarity), the loss function is as follows:

$$I_{i,j} = -log \frac{exp(sim(T_{N-K}, T_{N-K})/\tau)}{\Sigma_{k=1}^{2B} 1_{[k \neq i]} \, exp(sim(T_{N-K}, T_{N-K})/\tau)} \quad (1)$$

Here, $1_{[k \neq i]} \in 0, 1$ is an indicator function that takes the value of 1 when $k \neq i$. τ is the temperature parameter and B represents the batch size. The final loss value is calculated for all positive pairs $(i, j), (j, i)$ in the mini-batch.

3.3 Evaluation Method

The accuracy for a problem refers to the percentage of correctly predicted instances out of the total number of predictions, calculated as a real number between 0 and 100.

3.4 Experimental Results and Analysis

Compared the classification results when contrastive learning was applied (■) and not applied (■) to the representation vectors of input-output pairs. Two classification methods, linear and KNN, were used.

Table 1: Results of accuracy evaluation using KNN and linear search methods based on the presence or absence of contrastive learning in the ARC dataset.

Dataset	Classifier	Without CL (■)	With CL(■) SSL	With CL(■) SCL
ARC	KNN	17.31%	28.85%	32.93%
	Linear	23.56%	38.94%	40.14%
ConceptARC	KNN	11.36%	19.89%	16.48%
	Linear	20.11%	22.90%	22.90%

Through these experiments conducted on two distinct datasets, the application of contrastive learning consistently showed an improvement in performance, with an increase of up to 15 percentage points in KNN accuracy and a maximum of approximately 17 percentage points in linear probing accuracy, compared to methods without contrastive learning.

4. Conclusion

In our research, we initially explored the utilization of prior knowledge in artificial intelligence, particularly its significance in addressing ARC problems, and determined that information about problem types is a crucial form of such knowledge.
Notably, ConceptARC stands out as the only dataset currently classifying ARC problems by type. To bridge the gap created by the scarcity of similar datasets and the need for effective representation vector extraction for problem types, we proposed a pre-training method utilizing contrastive learning. Our experimental results confirmed that this approach significantly improves the classification of ARC problems by type, compared to methods that do not employ contrastive learning. Looking ahead, we anticipate that employing the problem types identified through this methodology as prior knowledge in future research will substantially enhance problem-solving capabilities