Enhancing Analogical Reasoning in the Abstraction and Reasoning Corpus via Model-Based RL

Jihwan Lee[1,†], Woochang Sim[1,†], Sejin Kim[1,*] and Sundong Kim[1,*]

[1] *Gwangju Institute of Science and Technology*

Abstract

This paper demonstrates that model-based reinforcement learning (model-based RL) is a suitable approach for the task of analogical reasoning. We hypothesize that model-based RL can solve analogical reasoning tasks more efficiently through the creation of internal models. To test this, we compared DreamerV3, a model-based RL method, with Proximal Policy Optimization, a model-free RL method, on the Abstraction and Reasoning Corpus (ARC) tasks. Our results indicate that model-based RL not only outperforms model-free RL in learning and generalizing from single tasks but also shows significant advantages in reasoning across similar tasks.

Keywords

Abstraction and Reasoning Corpus, Analogical Reasoning, Model-Based RL, DreamerV3, PPO

1. Introduction

The Abstraction and Reasoning Corpus (ARC) is known as a benchmark for evaluating abstraction and reasoning abilities [1]. Examples of ARC tasks can be seen in Fig. 1. To solve an ARC task, one needs the ability to abstract a common rule among given demos and apply this rule to a new test input grid to infer the corresponding grid [2]. Additionally, to solve untrained tasks, one must have the ability to discover various knowledge during the learning process and selectively apply this knowledge in the task-solving process [2].

This study interprets the two abilities required by ARC not as separate skills but as one concept of analogical reasoning. Analogical reasoning refers to the process of understanding new situations or solving problems based on existing knowledge that can be applied to different contexts or problems [3]. First, the process of solving an ARC task can be seen as solving a new problem (test input grid) based on the transformations from input grids to output grids in each demo (existing knowledge), which exactly fits the definition described above. Second, for a model trained on various ARC tasks to solve a new ARC task, it must understand the demos of the new ARC task (new situations) based on the solution processes (existing knowledge) applicable to the training tasks (different problems) and solve the task, which also aligns with the concept of

IARML@IJCAI'2024: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI'2024, August, 2024, Jeju, South Korea

[*] Corresponding authors.

[†] Equal contribution.

jihwan.lee@gm.gist.ac.kr (J. Lee); woochang@gm.gist.ac.kr (W. Sim); sejinkim@gist.ac.kr (S. Kim); sundong@gist.ac.kr (S. Kim)

CEUR Workshop Proceedings (CEUR-WS.org)

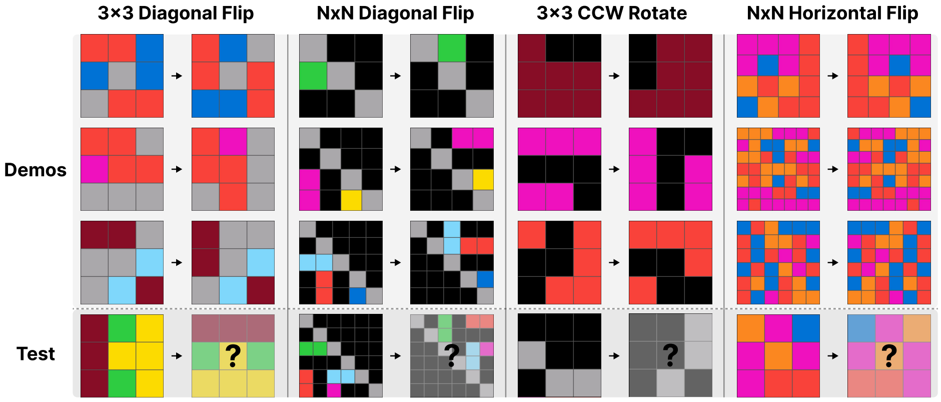


Figure 1: Each ARC task includes several demos and a test input. The objective is to identify the grid corresponding to the test input by applying a common transformation rule found across all demos.

analogical reasoning. Third, since the evaluation tasks in ARC are typically more complex and challenging than the training tasks [4], the ability to infer new problem-solving processes based on learned knowledge through analogical reasoning can enhance the performance on ARC.

In this study, noting the critical importance of analogical reasoning capabilities in solving ARC, we conducted reinforcement learning (RL) for training ARC tasks. Learning ARC with RL can offer the following potential benefits from an analogical reasoning perspective. First, RL can enhance its learning strategies through reward-based learning. In the process of learning ARC tasks, RL agents learn to maximize rewards across various tasks, thereby developing analogical reasoning capabilities. Second, RL can learn policies that effectively operate in new, yet similar environments based on experience. During the learning of various ARC tasks, RL agents can learn to apply rules learned in one scenario to another, a key element of analogical reasoning. Third, RL allows agents to produce their own data, enabling effective learning even without high-quality given data. In tasks like ARC, where data may be limited, RL agents can quickly infer patterns from a few demos and learn to apply these to the new input.

This study compared model-based RL and model-free RL to evaluate the efficiency of acquiring analogical reasoning abilities. The key difference between these two methodologies lies in the presence or absence of an internal environmental model. Model-based RL builds an internal model of the environment based on the agent's experiences and learns policies through predictable scenarios, which has the potential to significantly improve decision-making in complex situations. On the other hand, model-free RL learns to optimize rewards through direct interaction with the environment, allowing for the development of effective action policies without constructing an environmental model.

This study hypothesized that the internal model of model-based RL could aid in analogical reasoning. To test this hypothesis, we observed the learning processes of ARC tasks using DreamerV3 [5], a representative model-based RL algorithm, and Proximal Policy Optimization (PPO) [6], a general model-free RL algorithm, demonstrating the potential for RL agents to acquire analogical reasoning. Additionally, we evaluated the performance and learning efficiency of both algorithms by fine-tuning them on tasks similar to the pre-trained tasks and comparing the analogical reasoning capabilities they acquired.

2. Model-Based RL and Model-Free RL

RL has two main categories: model-based RL and model-free RL. Before conducting experiments to compare these categories from the perspective of abstraction reasoning, this paper briefly reviews the characteristics of each category and compares the algorithms that will be used in this experiment.

2.1. Model-Based RL

Model-based RL builds an internal model of the environment to predict various possibilities and establish more efficient policies based on these predictions. In problems requiring abstract thinking and pattern recognition like ARC, the model-based RL approach provides the ability to generalize to new situations that the agent has not experienced through its internal model. Representative algorithms utilizing the model-based RL approach include Dyna-Q [7], I2A [8], MBMF [9], and World Model [10]. Also, AlphaZero [11], which successfully surpassed human performance in board games such as Go, Chess, and Shogi based on Monte Carlo Tree Search (MCTS), is the most well-known. Recently, DreamerV3 [12], based on the World Model, has been the subject of extensive research.

A key feature of DreamerV3 is its ability to extract important features from input data and convert these into a latent representation, which is then used to learn a predictive model of the environment's dynamics. This model functions by taking the current state and actions as inputs and predicting the next state's representation and rewards. This approach allows for the simulation of future scenarios and the evaluation of various scenarios. In summary, DreamerV3 operates by repeating the following three processes: 1) collect data based on a random policy or the current policy, 2) update the agent's model based on the collected data, and 3) generate and evaluate virtual future scenarios using the updated model to improve the policy.

2.2. Model-Free RL

Model-free RL operates directly through interactions with the external environment, basing its action policies on reward information. This methodology utilizes only the agent's experiences without prior modeling of the environment, making it relatively simple to implement and fast to execute. However, in complex reasoning tasks like ARC, the model-free RL approach can be somewhat limited. Specifically, it cannot generalize in continuous or dynamic situations and has limitations in quickly learning changing patterns. These characteristics could constrain the flexible reasoning and rapid adaptation required in ARC. Representative algorithms of model-free RL include DQN [13], PPO [14], and SAC [15].

Among these, the Proximal Policy Optimization (PPO) algorithm is one of the most commonly used algorithms in RL, based on the policy gradient method. PPO is designed to achieve stable and efficient learning even in complex environments. Its process can be summarized similarly to DreamerV3 in the following two steps: 1) perform actions in the environment based on the current policy to collect data, and 2) incrementally improve the policy using the policy gradient technique based on the collected data.

3. Experiments

We conducted an experiment comparing the ARC task learning of DreamerV3 and PPO, which are representative algorithms of model-based RL and model-free RL, respectively, to demonstrate the differences in analogical reasoning capabilities between these methodologies. Through this experiment, we aimed to answer the following research questions (RQs):

RQ1. Can Model-Based RL Learn a Single Task? ($\mathcal{A} \rightarrow \mathcal{A}$)
RQ2. Can Model-Based RL Reason Tasks Similar to Pre-Trained Task? ($\mathcal{A} \rightarrow \mathcal{A}'$)
RQ3. Can Model-Based RL Reason Sub-Tasks of Pre-Trained Task? ($\mathcal{AB} \rightarrow \mathcal{A}$)
RQ4. Can Model-Based RL Learn Multiple Tasks Simultaneously? ($\mathcal{A}, \mathcal{B} \rightarrow \mathcal{A}$)
RQ5. Can Model-Based RL Reason Merged-Tasks of Pre-Trained Tasks? ($\mathcal{A}, \mathcal{B} \rightarrow \mathcal{AB}$)

We scope our paper to provide answers on RQ1 and RQ2. By addressing RQ1 and RQ2 through experiments, we aim to demonstrate the abilities of model-based RL in efficiently learning and applying knowledge across similar tasks. The following descriptions are the common setting of every experiment.

Action The vast action space of ARC is one of the biggest obstacles to the RL [2]. According to the ARCLE framework [16], action is defined as a combination of operation and selection; where operation means the type of action to be performed, and selection determines grids to which these actions are applied. We force strict restrictions on these operations and selections to weaken the challenge posed by the vast action space. In experiments, RL agents can select only five operations: Rotate90, Rotate270, FlipH, FlipV, and Submit [16], and selections are always fixed as the entire grid. These limitations influenced the selection of tasks.

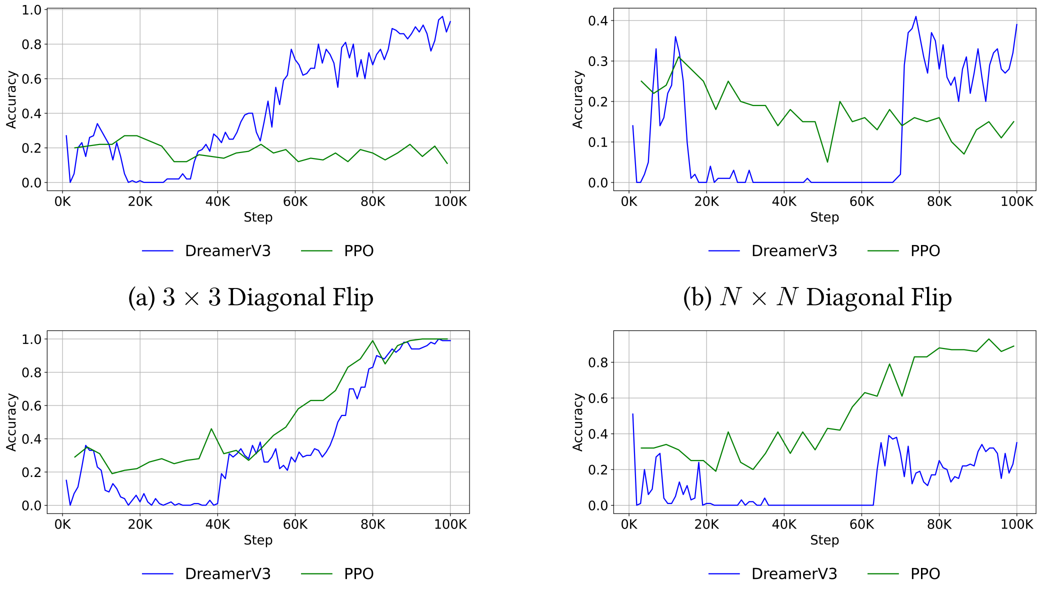
Task We found tasks from among the 400 training tasks that could be solved with the restricted operations and selections of actions, and 7 tasks satisfied this condition. Among these tasks, we selected 4 tasks for our experiment as shown in Fig. 1. First, we selected tasks that need diagonal flipping on a 3×3 grid or $N \times N$ grids. These tasks need the right combination of actions (rotate and flip) before submission. Thus, we thought these tasks were good for measuring analogical reasoning ability. Next, we chose tasks that need to rotate a 3×3 grid and flip horizontally on $N \times N$ grids. These tasks need only one action except submit, so analogical reasoning ability are not necessary for agents.

Reward We designed the reward based on a sparse reward such that an agent gets a reward of 1000 if it does the submit action and the state matches the correct grid. Additionally, even if the submit action is not executed, reaching the correct grid grants an extra reward of 1. Although sparse rewards can present challenges in learning tasks with a wide search space, we overcame these difficulties through simple tasks and restricted action spaces. Furthermore, by giving a small reward for just finding the correct grid, the RL agent can learn the importance of achieving the goal state, and by offering a large reward for the submit action, we prevented the agent from repeatedly reaching the correct grid without submitting in a single episode.

Metric In all experiments, we measured the data efficiency of learning through the accuracy relative to the number of environment steps used during training. We decided to use environment steps to ensure consistency with the experimental settings of DreamerV3 [12], as we wanted to monitor data efficiency throughout the learning process. Additionally, we employed the pass@3 method used in ARC to measure accuracy [2]. Pass@3 grants three submission opportunities per episode, and an episode is considered successful if the correct answer is found within three attempts. If three incorrect submissions are made or the predetermined episode length (50) is exceeded, the attempt is judged unsuccessful.

Training All tasks used in the experiments were trained using the RL environment for ARC, ARCLE [16]. Each task was originally composed of a few demos and one test input; however, there was a concern that this could lead to overfitting during the learning process. Therefore, we augmented the training with 1,000 demos and used an additional 100 test inputs for evaluating the model. Furthermore, when training a single task, the agent was trained over 100,000 environment steps, whereas for fine-tuning a new task on a pre-trained model, the agent was trained for 50,000 environment steps.

3.1. Can Model-Based RL Learn a Single Task? ($\mathcal{A} \rightarrow \mathcal{A}$)



(a) 3×3 Diagonal Flip

(b) $N \times N$ Diagonal Flip

(c) 3×3 CCW Rotate

(d) $N \times N$ Horizontal Flip

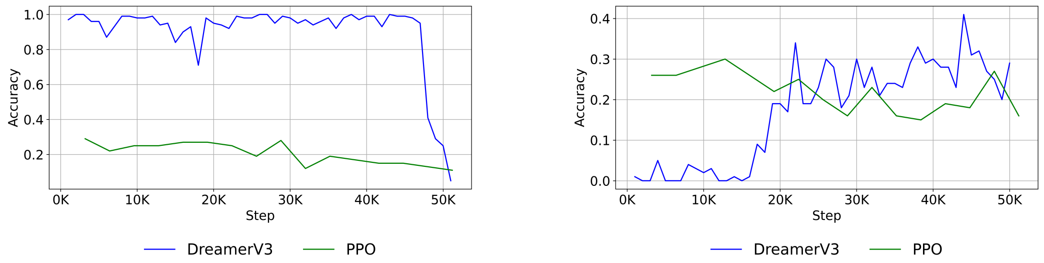
Figure 2: Performance of agents on four single ARC tasks with different RL algorithms. The above two results show that the model-based RL agent learned better on analogical reasoning tasks. The below two results show that the model-free RL agent could be better on simple tasks. Additionally, an interesting common result was shown in the learning curve of model-based RL: there always occurs an interval in the middle of learning where accuracy drops to 0. We argue that this interval is where model-based RL learns concepts for analogical reasoning.

The first experiment focused on a single ARC task requiring analogical reasoning, where the objective was to learn the common rules from given demo pairs and apply these rules effectively to a test input. This experiment involved training on four ARC tasks, each illustrated in Fig. 1, using the DreamerV3 and PPO algorithms. Tasks requiring multiple actions to solve, such as Diagonal Flipping, demand analogical reasoning, whereas tasks that can be solved with a single action, such as CCW Rotation and Horizontal Flipping, are less complex. DreamerV3, a model-based RL algorithm, exhibited superior performance in Diagonal Flip tasks, which involve more complex reasoning. In contrast, PPO, recognized for its stable learning capabilities, performed comparably or even better in simpler tasks.

It's noteworthy that DreamerV3 achieved 100% performance on uniformly sized 3×3 tasks but only managed about 40% on tasks with varying sizes. Conversely, PPO's performance remained consistent regardless of grid size and was influenced solely by task difficulty. This suggests a performance drop in DreamerV3 during the encoding processes.

A common interval of zero performance was observed in DreamerV3's training across various tasks, followed by rapid performance improvement. This pattern suggests that DreamerV3 is not merely learning simple problem patterns but is developing a deeper understanding of concepts essential for analogical reasoning. Initially, DreamerV3 demonstrated about 40% effectiveness due to learning straightforward patterns. Later, the agent recognized that this approach was insufficient for fully solving tasks, prompting it to explore and learn from various trials. Consequently, DreamerV3 displayed significant performance improvements in later learning stages, outperforming earlier results. These observations imply that DreamerV3's latent representations capture conceptual content that enhances learning efficiency and illustrates the potential for analogical reasoning within model-based RL frameworks.

3.2. Can Model-Based RL Reason Tasks Similar to Pre-Trained Task? ($\mathcal{A} \rightarrow \mathcal{A}'$)



(a) 3×3 to $N \times N$ Diagonal Flip

(b) $N \times N$ to 3×3 Diagonal Flip

Figure 3: Comparing the performance of agents between DreamerV3 and PPO on two single ARC tasks with a pre-trained model about a similar task. Understandably, PPO did not gain any benefit from fine-tuning due to the low performance of the pre-trained model. In contrast, DreamerV3 showed very high performance when adapting from a pre-trained model that had performed well in the 3×3 Diagonal Flip task. However, when utilizing a pre-trained model with poor performance, DreamerV3 also displayed lower initial learning efficiency than when no pre-trained model was used. Lastly, at the end of the experiment where fine-tuning was successful, a sudden drop in performance occurred, which is presumed to be the same phenomenon as the interval in previous experiments.

The second experiment evaluated the learning efficiency of an agent when facing a task similar to one it had previously trained on. For instance, in Fig. 3a, we observed the performance of diagonal flip tasks across various grid sizes for agents pre-trained on a 3×3 diagonal flip task. Conversely, in Fig. 3b, we reversed the roles of the pre-trained and adaptation tasks to evaluate both agents' performances.

In both scenarios, PPO consistently displayed a performance level of around 20%. This outcome was anticipated, considering that the agent's performance on pre-trained tasks had similarly been around 20%, as evidenced in previous experiments (Fig. 2a and Fig. 2b). The insufficient performance of the pre-trained model likely inhibited any positive influence on the adaptation process, resulting in no notable performance gains compared to the non-pre-trained scenario.

An interesting finding from the DreamerV3 results was the direct impact of the pre-trained model's performance on adaptation. For example, while the model without pre-training achieved about 40% performance on the same task, the pre-trained model approached 100%, mirroring the 3×3 Diagonal Flip task's success previously encountered. This pattern was further supported by the performance contrasts observed between Fig. 2a and Fig. 3b. Here, a task that once achieved 100% performance dropped to around 40% after pre-training, reflecting the outcomes seen with the $N \times N$ Diagonal Flip tasks. These outcomes illustrate that DreamerV3 tailored its problem-solving strategies based on insights drawn from pre-trained tasks, suggesting that better results could have been obtained if adaptation in Fig. 3b had also been initiated from a higher-performing pre-trained model.

Towards the latter part of training in Fig. 3a, a sudden decline in performance was noted. While the exact cause of this drop cannot be definitively established, we hypothesize that it might be due to the same type of interval observed in previous results (Fig. 2). Although the experiment's step limit prevented precise verification, it seems probable that the DreamerV3 agent was undergoing an interval similar to earlier tests, attempting to learn new concepts. Further experimentation beyond 50,000 steps and additional theoretical analysis are required to explore this phenomenon more deeply, which could further substantiate DreamerV3's capability for analogical reasoning.

4. Discussion

Restriction of Action Space Our study examined the analogical reasoning abilities of reinforcement learning (RL) algorithms and their ability to apply learned concepts. In the ARCLE, an agent's actions consist of operation and selection, requiring the policy to adeptly handle both components. However, the challenge of learning the selection process significantly increases the complexity of the tasks. To mitigate this, we simplified the action space by providing selections through an oracle in this experiment. This adjustment allowed the agents to focus more on mastering operations without the added difficulty of selection, streamlining the learning process. Future research should explore the possibility of incorporating the selection process more autonomously within the reasoning tasks. This would test the agent's ability to handle the complete process of decision-making, potentially leading to more analogical reasoning ability for agents.

Restriction of Tasks This paper focused on learning simple tasks involving just one or two operations, such as diagonal flipping tasks, a Rotating task, and a horizontal flipping task. Future research should explore additional challenges such as the Rotate and the Flip task, which could provide valuable insights into the model's ability to generalize and reason about sub-components of learned tasks. It is also crucial to assess the efficiency of learning multiple related actions (multi-tasking) simultaneously. To overcome this challenge, previously learned knowledge must be retained while new knowledge is acquired. In other words, it is essential to be robust against catastrophic forgetting. Should such an agent be developed, two types of experiments will be essential: 1) The agent should be trained on a large and diverse set of tasks to verify the absence of catastrophic forgetting. 2) The agent should be trained on a wide range of tasks with varying difficulties, followed by an evaluation of its capability to adapt to untrained tasks.

Applying Meta-Learning In addressing the ARC tasks, the ability to apply learned concepts to entirely new contexts is essential. Meta-learning, or 'learning to learn,' can be considered to adapt quickly to new tasks. Future studies should focus on developing meta-learning frameworks that can efficiently abstract underlying task structures, allowing for rapid generalization across varying problem settings. A deeper exploration into meta-learning could utilize frameworks such as Model-Agnostic Meta-Learning (MAML) [17] which has shown promise in various domains. Implementing and refining meta-learning techniques like MAML could lead to breakthroughs in developing AI models that can learn not only untrained tasks more efficiently but also their knowledge to solve untrained tasks.

Applying Transfer Learning Transfer learning can significantly enhance the model's ability to utilize knowledge acquired in one context and apply it to different yet related tasks. This process often involves the use of deep neural networks, specifically their ability to approximate and adapt policies through layers that capture generalizable features, which are crucial for the successful transfer of knowledge across tasks within the same domain or between different domains [18]. Conducting such studies would provide deeper insights into the flexible application of acquired knowledge, a critical aspect of analogical reasoning. These investigations will pave the way for creating adaptable and efficient learning systems that thrive in dynamically changing environments.

Theoretical Analysis of Model-Based RL Finally, our findings indicate that model-based RL can facilitate more efficient learning in analogical reasoning tasks compared to model-free approaches. The inherent capabilities of model-based methods to infer and generalize from limited data were particularly beneficial. However, our study also highlights the need for further investigation into the specific mechanisms through which these models store and retrieve task-specific patterns. A deeper understanding of these processes could inform the development of more robust model-based systems, enhancing their capability to handle a wider array of complex reasoning tasks.

5. Related Works

The winner of the ARC Challenge [19] utilized a total of 142 domain-specific languages (DSLs) to combine various transformed images [20]. The ARC Challenge later expanded into ARCathon [21], and the ARCathon 2022 winner's approach involved exploring using 166 DSLs that included more complex and diverse features [22]. This attempted to challenge the broad generalization of DSLs; however, it also introduced limitations due to the inclusion of very infrequently used features, making the DSLs too complex. Compared to humans can solve about 80% of ARC evaluation tasks [23], DSL-based search algorithms have shown performances around 30-40% [4, 21]. However, these approaches, being fundamentally simple search-based, have inherent weaknesses in complex tasks and are prone to overfitting to specific tasks defined by artificially designed DSLs, making them difficult to apply or generalize to other tasks. This fundamental limitation may render them unsuitable for analogical reasoning.

Subsequent studies have attempted to solve tasks using neural networks. Some research efforts were based on program synthesis, progressively recognizing increasingly complex patterns, while other studies have tried to learn the complexities of ARC through the computational abilities of Large Language Models (LLMs). However, studies without DSLs have shown relatively low performance. Even the most recent studies based on program synthesis [24] or LLMs [25] have only achieved performances of 6.5–6.75% on untrained tasks [26]. Recently, there was interesting research utilizing human-solving processes [27], but this study did not contain the performance on untrained tasks. The low performance of these studies is attributed to the following reasons: In the case of program synthesis, the method of combining all possessed knowledge to acquire new knowledge leads to poor learning efficiency [28], and LLMs are known to be weak in incremental reasoning [29]. These limitations can be particularly harmful in evaluation tasks, which are known to be comparatively more complex and difficult than training tasks [4]. Additionally, the research utilizing human-solving processes has a risk to untrained tasks due to their reliance on offline learning from given solutions [30].

Ensemble-based research that combines the previous approaches has also been announced [4, 26]. These methods were expected to compensate for the weaknesses of existing approaches through the ensemble. An ensemble-based study combining program synthesis and LLMs showed a performance of 14.75% [26], and the approach of the ARCathon 2023 winner exhibited about 33% performance, significantly surpassing previous studies. However, the performance of ensemble-based research that combined DSL, program synthesis, and LLMs was only 40.25% [4], which is just 0.25% higher than the performance of a baseline that used only DSL. This result means that just one more task out of 400 evaluation tasks was solved. Ultimately, the experimental results of these ensemble-based studies further highlighted the limitation that the presence of program synthesis and LLMs does not significantly impact performance compared to DSL.

Meanwhile, there are some attempts to apply reinforcement learning (RL) algorithms in a limited way based on program synthesis and LLMs methods [26, 31]. The limited application of RL algorithms can be attributed to factors such as the absence of appropriate rewards, high-dimensional complex states, and extensive search spaces, but fundamentally, the absence of an environment for ARC was the main reason. However, an environment for training ARC tasks [16] has been developed recently. Furthermore, this research demonstrated successful training of an ARC task using PPO, suggesting the potential applicability of various RL models.

6. Conclusion

In this paper, we compared analogical reasoning abilities between model-based and model-free RL, using the ARC to demonstrate the effectiveness of model-based RL. This approach has shown superior learning efficiency and adaptability in tasks compared to model-free methods like PPO. Specifically, DreamerV3 outperformed in tasks similar to those it had previously solved, displaying a remarkable ability to generalize across various task dimensions.

The observed performance dips followed by rapid recoveries during DreamerV3's training are thought to represent periods of conceptual consolidation. These intervals align with theoretical expectations of analogical reasoning in AI systems and highlight DreamerV3's potential to manage complex cognitive tasks. Future research should further explore these learning dynamics to better understand the mechanisms that enable such advanced reasoning. This will help enhance RL agents capable of analogical reasoning and adaptation in dynamic environments, reducing the gap between the reasoning ability of humans and RL agents.

Acknowledgments

This work was supported by the IITP (No. 2019-0-01842, RS-2023-00216011), the National Research Foundation (RS-2023-00240062), and the GIST (AI-based Research Scientist Project) funded by the Ministry of Science and ICT, Korea.

References

[1] M. Małkiński, J. Mańdziuk, A Review of Emerging Research Directions in Abstract Visual Reasoning, Information Fusion 91 (2023) 713–736.

[2] F. Chollet, On the Measure of Intelligence, arXiv:1911.01547 (2019).

[3] D. R. Hofstadter, Gödel, Escher, Bach: An Eternal Golden Braid, Basic books, 1979.

[4] M. Bober-Irizar, S. Banerjee, Neural Networks for Abstraction and Reasoning: Towards Broad Generalization in Machines, arXiv:2402.03507 (2024).

[5] D. Hafner, J. Pasukonis, J. Ba, T. Lillicrap, Mastering Diverse Domains through World Models, arXiv:2301.04104 (2023).

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization algorithms, arXiv:1707.06347 (2017).

[7] R. S. Sutton, Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming, in: ICML, 1990, pp. 216–224.

[8] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, et al., Imagination-Augmented Agents for Deep Reinforcement Learning, in: NeurIPS, 2017.

[9] A. Nagabandi, G. Kahn, R. S. Fearing, S. Levine, Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning, in: ICRA, 2018, pp. 7559–7566.

[10] D. Ha, J. Schmidhuber, World Models, arXiv preprint arXiv:1803.10122 (2018).

[11] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al., Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, arXiv:1712.01815 (2017).

[12] D. Hafner, J. Pasukonis, J. Ba, T. Lillicrap, Mastering Diverse Domains through World Models, arXiv preprint arXiv:2301.04104 (2023).

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning, arXiv preprint arXiv:1312.5602 (2013).

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, arXiv:1707.06347 (2017).

[15] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, in: ICML, 2018, pp. 1861–1870.

[16] H. Lee, S. Kim, S. Lee, S. Hwang, J. Lee, B.-J. Lee, S. Kim, ARCLE: The Abstraction and Reasoning Corpus Learning Environment for Reinforcement Learning, in: CoLLAs, 2024. URL: https://openreview.net/forum?id=EE33zbaLwa.

[17] C. Finn, P. Abbeel, S. Levine, Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, in: ICML, 2017, pp. 1126–1135.

[18] M. E. Taylor, P. Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, Journal of Machine Learning Research 10 (2009).

[19] Kaggle, Abstraction and Reasoning Challenge, 2020. URL: https://www.kaggle.com/c/abstraction-and-reasoning-challenge/.

[20] J. S. Wind, DSL Solution to the ARC Challenge, 2020. URL: https://github.com/top-quarks/ARC-solution/.

[21] Lab42, ARCathon: Global AI Competition, 2024. URL: https://lab42.global/arcathon/.

[22] M. Hodel, Domain Specific Language for the Abstraction and Reasoning Corpus, 2023. URL: https://github.com/michaelhodel/arc-dsl.

[23] A. Johnson, W. K. Vong, B. M. Lake, T. M. Gureckis, Fast and Flexible: Human Program Induction in Abstract Reasoning Tasks, in: CogSci, 2021, pp. 2471–2477.

[24] J. Ainooson, D. Sanyal, J. P. Michelson, Y. Yang, M. Kunda, A Neurodiversity-Inspired Solver for the Abstraction & Reasoning Corpus (ARC) Using Visual Imagery and Program Synthesis, arXiv:2302.09425 (2023).

[25] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, A. Zeng, Large Language Models as General Pattern Machines, arXiv:2307.04721 (2023).

[26] N. Butt, B. Manczak, A. Wiggers, C. Rainone, D. Zhang, M. Defferrard, T. Cohen, CodeIt: Self-Improving Language Models with Prioritized Hindsight Replay, arXiv:2402.04858 (2024).

[27] J. Park, J. Im, S. Hwang, M. Lim, S. Ualibekova, S. Kim, S. Kim, Unraveling the ARC Puzzle: Mimicking Human Solutions with Object-Centric Decision Transformer, in: ICML ILHF Workshop, 2023.

[28] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, J. B. Tenenbaum, Dreamcoder: Bootstrapping Inductive Program Synthesis with Wake-Sleep Library Learning, in: PLDI, 2021, pp. 835–850.

[29] S. Lee, W. Sim, D. Shin, S. Hwang, W. Seo, J. Park, S. Lee, S. Kim, S. Kim, Reasoning Abilities of Large Language Models: In-Depth Analysis on the Abstraction and Reasoning Corpus, arXiv:2403.11793 (2024).

[30] J. C. M. Tan, M. Motani, Large Language Model (LLM) as a System of Multiple Expert Agents: An Approach to Solve the Abstraction and Reasoning Corpus (Arc) Challenge, arXiv:2310.05146 (2023).

[31] S. Alford, A. Gandhi, A. Rangamani, A. Banburski, T. Wang, S. Dandekar, J. Chin, T. Poggio, P. Chin, Neural-Guided, Bidirectional Program Search for Abstraction and Reasoning, in: Complex Networks, 2021, pp. 657–668.