Plan-and-Execute: LLM-Guided Reinforcement Learning with Cross-Modal Fusion for Long-Sequence Decision Making

Extended Abstract

Yadong Li
Nanjing University of Science and Technology
Nanjing, China
Zaozhuang University
Zaozhuang, China
liyadong@njust.edu.cn

Tong Zhang
Nanjing University of Science and Technology
Nanjing, China
tong.zhang@njust.edu.cn

Zhen Cui*
School of Artificial Intelligence,
Beijing Normal University
Beijing, China
zhen.cui@bnu.edu.cn

ABSTRACT

We introduce **Pl**an-and-**Ex**ecute (PLEX), a novel framework that synergies the abstract reasoning capabilities of large language models (LLMs) with the grounded reinforcement learning (RL). In this architecture, an LLM serves as a dynamic planner, iteratively decomposing complex language instructions into structured subgoal sequences. A dedicated RL agent, leveraging cross-modal attention mechanism, then executes these subgoals by fusing language instructions with high-dimensional visual observations to learn and optimize its decision-making policies. This hierarchical coordination enables our agent to master long-horizon tasks through a tight loop of reasoning and grounded interaction. Comprehensive evaluation in the MiniGrid and MiniHack environment confirms that PLEX achieves a significant performance improvement over existing methods across diverse scenarios. The PLEX exhibits superior sample efficiency, particularly in complex, long-horizon tasks that require sustained reasoning and action sequences.

KEYWORDS

Large Language Models; Iterative task planner; Ground Reinforcement Learning; Multi-modal Cross-Attention

ACM Reference Format:
Yadong Li, Tong Zhang, and Zhen Cui*. 2026. Plan-and-Execute: LLM-Guided Reinforcement Learning with Cross-Modal Fusion for Long-Sequence Decision Making: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/http://doi.org/10.65109/WKWD5060

1 INTRODUCTION

Deep reinforcement learning (RL) [10] has achieved remarkable progress in recent years, demonstrating human-level or even superior performance in controlled environments with well-defined reward signals, ranging from game playing [2, 8, 11, 14] to robotic control [3, 5, 7, 15]. However, developing autonomous agents capable of natural collaboration with humans remains a central challenge in artificial intelligence. This goal not only requires agents to perform complex tasks but also necessitates their ability to understand and respond to natural language instructions, enabling flexible decision-making in open, dynamic, and highly uncertain environments.

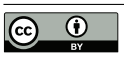
While existing solutions show promise, recent advances in large language models (LLMs)[1] provide new opportunities to address these challenges by enabling strong semantic understanding, task planning, and step-by-step reasoning from natural language instructions [13]. Building on these capabilities, several LLM-based agent frameworks have been proposed. For example, ELLM [6] leverages LLM-generated goals to guide the training and fine-tuning of reinforcement learning agents, and Voyager [12] uses an LLM to generate exploration curricula and corresponding skill code, achieving impressive performance in Minecraft. Despite these advances, existing methods still rely on manually designed low-level policies, limiting true end-to-end control, and are largely evaluated in closed-game environments, with limited validation in complex open-world scenarios.

To address these challenges, we propose PLEX (**Pl**an-and-**Ex**ecute), an interactive planning architecture that synergizes large language models with reinforcement learning. In this framework, an LLM acts as a dynamic planner, translating natural language instructions into logically structured and temporally constrained subtask sequences. This design directly addresses the challenge of long-horizon reasoning by providing explicit subgoals that mitigate sparse reward issues and establish coherent task progression.

2 METHODS

We propose an interactive plan-and-execute paradigm called PLEX, which integrates LLMs and RL methods. In this framework, the LLM serves as the core "planner" for task planning, enabling semantic understanding and decomposition of natural language instructions into sequences of subtasks with strict logical order and temporal constraints. Simultaneously, the RL module acts as the "executor" of the agent, responsible for learning and optimizing the execution policies of each subtask.

For the processing of the two modalities, the observed image $I \in \mathbb{R}^{C\times H\times W}$ is sliced into patches and flattened into $v \in \mathbb{R}^{N\times(P^2\times C)}$, where (P, P) is the patch resolution, and $N = HW/p^2$ represents



Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/http://doi.org/10.65109/WKWD5060

the total number of patches. Subsequently, a linear projection $V \in \mathbb{R}^{(P^2 \times C) \times H}$ is applied to v, which is then augmented with positional embeddings $V^{pos} \in \mathbb{R}^{(N+1) \times H}$, resulting in the final embedded representation $\bar{v} \in \mathbb{R}^{N \times H}$.

$$\bar{v} = [v_1 V; ...; v_N V] + V^{pos} \tag{1}$$

For the language instruction g, it is processed using sentence embedding, resulting in the corresponding feature vector $\bar{g}$. This approach enables a more accurate capture of the overall semantic meaning, reducing the bias caused by differences in linguistic expression and thereby enhancing the agent's understanding of instructions and its generalization capability.

To enable the agent to jointly attend to information from different representation subspaces, we employ a multi-head cross-attention mechanism. The language embedding $\bar{g}$ serves as the Query to attend over the visual feature sequence v. Specifically, the input projections are first split into h heads. For each head i, we compute a scaled dot-product attention:

$$\begin{aligned} \text{head}_i &= \text{Attention}(\bar{g}W_Q^i, \bar{v}W_K^i, \bar{v}W_V^i) \\ &= \text{softmax}\left(\frac{\bar{g}W_Q^i(\bar{v}W_K^i)^T}{\sqrt{d_k}}\right)\bar{v}W_V^i \end{aligned} \tag{2}$$

where the projections are parameter matrices $W_Q^i \in \mathbb{R}^{d \times d_k}$, $W_K^i \in \mathbb{R}^{H \times d_k}$, $W_V^i \in \mathbb{R}^{H \times d_v}$, and $d_k = d_v = d_{\text{model}}/h$.

The outputs of all heads are then concatenated and projected to yield the final state representation:

$$s = W_O[\text{head}_1; \ldots; \text{head}_h] \tag{3}$$

where $W_O \in \mathbb{R}^{(h \cdot d_v) \times d_{\text{model}}}$ is the output projection matrix. This multi-head design allows the model to capture diverse relationships between the language instruction and different visual regions simultaneously, leading to a more robust and informative state representation s for policy learning.

3 EXPERIMENTS AND RESULTS

We conducted experiments on the Minigrid [4] and Minihack [9] benchmarks to validate the performance of PLEX. We first evaluated the performance of PLEX and baseline algorithms in for MiniGrid scenarios, the experimental results of compared performance are shown in Figure. 1.

By integrating LLMs with RL, PLEX effectively addresses long-horizon tasks. In challenging environments such as KCmedium, and KChard, traditional RL algorithms like PPO and A2C struggle to learn effective policies. In contrast, PLEX demonstrates superior performance. Notably, curiosity-driven methods like NovelD can learn effective strategies in many scenarios. In simpler tasks, their performance is comparable to that of PLEX. However, in complex environments (KChard), NovelD's reliance on intrinsic rewards limits its sample efficiency. In contrast, PLEX, guided by its planner, quickly learns core strategies and reduces ineffective exploration steps, achieving the highest average reward. This success stems from the LLM's strong reasoning capabilities, which extract key information from task descriptions and combine it with the agent's state to decompose complex tasks into executable subtasks. This

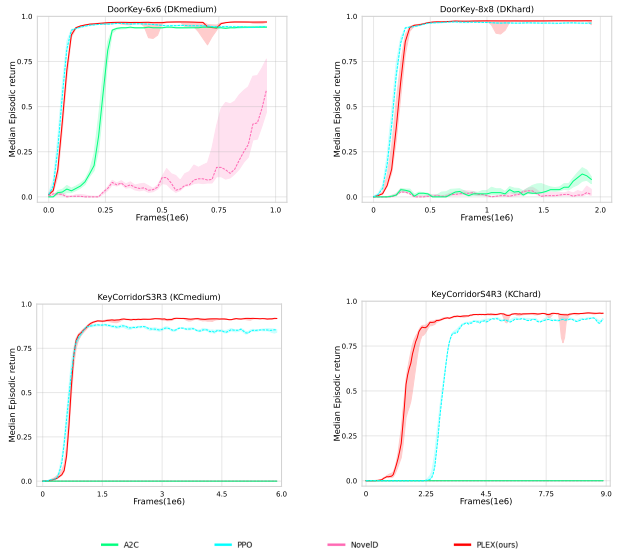


Figure 1: The results of four algorithms in MiniGrid.

hierarchical approach improves learning efficiency and enhances adaptability in diverse and challenging environments.

Consider that A2C and PPO exhibit similar performance, we focus on comparing PPO with NovelD and PLEX in the MiniHack environments, the results across four tasks are presented in Table 1.

PLEX consistently outperforms both PPO and NovelD across challenging task settings. While PPO succeeds in simpler configurations such as LC-medium—where the objective is reduced to goal navigation once a key item is pre-equipped—it fails to develop coherent strategies in scenarios requiring sequenced skill execution (e.g., potion retrieval, wand activation, and kill the monster). Notably, PLEX sustains robust performance even in the demanding Wod-hard environment, where NovelD achieves no meaningful progress, underscoring PLEX's capacity to integrate reasoning with high-dimensional visual inputs under sparse reward conditions.

Table 1: Comparative Performance Evaluation in MiniHack Environment.

	LCmedium	LChard	WoDmedium	WoDhard
PPO	0.9	0.42	0.01	0.0
NovelD	0.91	0.54	0.15	0.1
PLEX	0.91	0.71	0.68	0.62

4 CONCLUSION

In this work, we propose Plan-and-Execute, a novel framework that integrates large language models with reinforcement learning to tackle long-sequence task with sparse rewards. The framework employs an LLM-based planner to iteratively decompose natural language instructions into logically structured subtask sequences, while a multi-head cross-attention mechanism dynamically grounds these instructions in visual observations to guide policy learning. This hierarchical design significantly enhances the agent's ability to interpret and execute language-guided tasks, while improving generalization in open-domain settings.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[3] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[4] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. 2024. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems* 36 (2024).

[5] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. 2023. Palm-e: An embodied multimodal language model. (2023).

[6] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*. PMLR, 8657–8677.

[7] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. 2023. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9493–9500.

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[9] Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. 2021. Minihack the planet: A sandbox for open-ended reinforcement learning research. *arXiv preprint arXiv:2109.13202* (2021).

[10] Richard S Sutton. 2018. Reinforcement learning: An introduction. *A Bradford Book* (2018).

[11] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[12] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).

[13] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[14] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*. 3620–3624.

[15] Qinqing Zheng, Amy Zhang, and Aditya Grover. 2022. Online decision transformer. In *international conference on machine learning*. PMLR, 27042–27059.