Towards Automated Reinforcement Learning: Applications and Prospects of Large Language Models as Cognitive Components in the Full RL Lifecycle

Shenghao Yuan

{Shenghao.Yuan23@student.xjtlu.edu.cn}

Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, 25000, China

Abstract: Deep reinforcement learning (DRL) has made significant progress in recent years. However, it still faces multiple bottlenecks in real-world applications. Agents suffer from low sample efficiency, and designing effective reward functions remains very difficult. Furthermore, traditional DRL lacks general cognitive abilities. Recent advances in large language models (LLMs) provide a promising solution. By utilizing the strong coding and reasoning skills of LLMs, researchers can overcome many of these limitations. Addressing the bottlenecks of low sample efficiency and difficult reward design in real-world deep reinforcement learning applications, this paper constructs a taxonomic framework of "large language model-driven reinforcement learning throughout its entire lifecycle," systematically exploring the proxy mechanism of large models for human experts in four stages: environment, reward, decision-making, and collaboration. This study finds that the code generation capabilities of large models enable the automated construction of simulated scenarios and dense rewards;the "hierarchical planning" effectively balances the common-sense nature and real-time nature of decision-making, while the natural language interface significantly enhances the interpretability of multi-agent collaboration. This indicates that the deep integration of large models and reinforcement learning achieves a closed loop of cognition and action, providing a key technical path for building artificial general intelligence.

Keywords: Large language model; reinforcement learning; full lifecycle; intelligent agent; artificial general intelligence

1. Introduction

In recent years, reinforcement learning has yielded remarkable results and is widely regarded as one of the core paths to artificial general intelligence (AGI) [1, 2]. Although it has excelled in tasks like Go and the control of robots, deep reinforcement learning (DRL) suffers four main limitations in progressing to open world tasks; inefficiency to samples, reward design challenges, lack of generalization capability, and cannot be scaled to human supervision. One of them is low sample efficiency that relates to the assumption that the agent does not possess any prior knowledge, and typically he/she must begin on Tabula Rasa and perform gigantic trial and error.

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365229

According to Pang et al. this inefficiency results in incredibly high training time and cost of computing power, which dramatically prevents the application of RL in the real-world problems of complexity like autonomous driving [3]. Second, complexity of structural rewards design is based on the fact that reward mechanism of complex tasks is difficult to define. Correlated studies indicate that in environments where the reward is sparse, agents are more likely to become confused as they cannot obtain feedback instantly, and this factor directly causes the exploration failure to take place [4, 5]. In addition, the inability to make generalizations reduces the flexibility of the model in non-stationary situations. Traditional models are prone to overfitting certain training realities, and failing to provide more general human knowledge, which in their study by Sha et al. This allows the models to struggle in trying to comprehend and process long-tail traffic situations they have never been exposed to, as do humans [6]. Lastly, the human supervision bottleneck of scalability is also starting to be more evident. The scarcity of energy of human experts, high cost of annotation, the occurrence of subjective bias makes traditional approaches based on manual feedback no longer adequate to satisfy the demands of the modern machine learning of so many high-quality feedback signals. Even though the scholarly community has started focusing on the use of Large Language Models in conjunction with RL, current work on this topic is highly fragmented. The current literature is mostly devoted to the use of LLM in particular sub-tasks (such as simple code generation or communication) and does not provide a more systematic view of the way to use LLM to re-organize the entire engineering lifecycle of reinforcement learning [7, 8].

The objective of the paper is to overcome the disjointed constraints of the available literature and critically delve into the state of the art in terms of Large Language Model-Enhanced Reinforcement Learning (LLM-Enhanced RL). In contrast to RLHF, where human feedback is used to optimize the model, the given study concentrates on the role of LLM in using prior knowledge as a human expert in the whole process of reinforcement learning. The study will be based on three main dimensions: the first is theoretical construction which develops the full lifecycle, including the environment, reward, and decision-making and collaboration; the second is mechanism revelation, to study the working mechanism of LLM and whether it replaces human intervention with a model one, and finally, the paradigm verification which explores the paradigm alteration where the human in the loop is replaced by a model in the loop and verifies its vitality in reducing human intervention and enhancing system automation and generalization.

2. Research Methods and Technical Approach

2.1 Theoretical Framework Overview

This study offers a conceptualization grounded in the full lifecycle view towards providing a comprehensive elaboration of how the process of reinforcing learning can be recreated by means of large language models. RL cycle consists of four sequential and interlocking phases related to environment and task construction, reward construction and assessment, policy search and implementation, and multi-agent interaction. This paper suggests a theoretical approach of full life cycle reconstruction as depicted in Figure 1. Whether compared to the conventional research that concentrates on just one step of the model training procedure, the given framework incorporates the Large Language Model (LLM) into the overall closed loop of reinforcement

learning (RL) between building and deploying it.

In this closed loop, in particular, at the environment building phase the LLM plays the role of a world architect, responding to the vague natural language requirements by writing and executing structured simulation environment code; In the reward engineering phase the LLM plays the role of a Reward Shaper, automatically generating and optimizing dense reward functions to solve the sparse feedback problem. During the decision-making step, the LLM is the "commander, " whose common sense guides the high level, planning, and implementation of the underlying plans, which are the RL strategies. The next stage is Collaboration stage where the role of LLM is that of a "communicator, " it works with the natural language interfaces to rebuild the communication and cooperation among several agents.

Through this framework, the general cognitive ability of LLM and the trial-and-error exploration ability of RL are deeply coupled, together forming a general intelligent system with self-evolution capabilities.

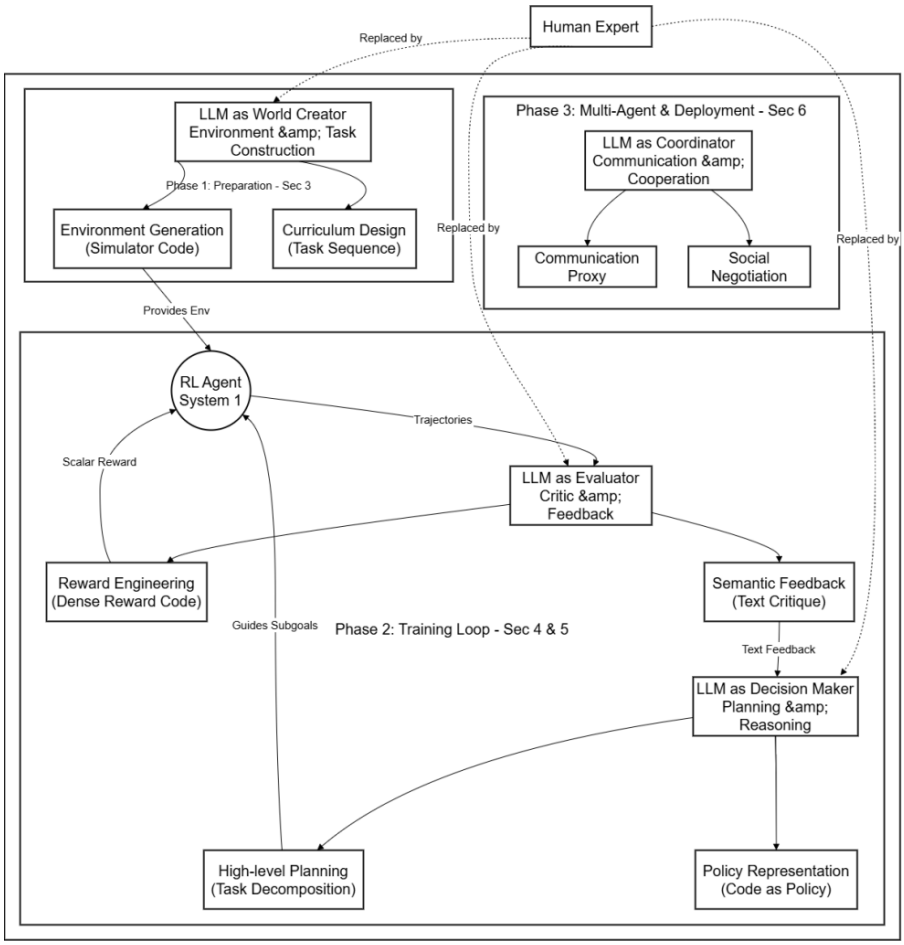

Fig. 1. Schematic diagram of the LLM-in-the-loop full lifecycle (Photo/Picture credit : Original).

2.2 Automation of environment building and task definition

Traditional reinforcement learning relies on manually constructed simulation environments, which is costly and difficult to adapt to changes. LLM can transform vague natural language instructions into structured environment configuration code, enabling "what you think is what you get" environment generation. This paper examines the research of EnvGen[9]. As shown in Figure 2, LLM is used as a "world architect". It can not only automatically generate training levels of different difficulties (such as maze layouts and obstacle positions) according to the teaching syllabus, but also dynamically adjust environmental parameters for targeted training by analyzing the failure trajectory of the agent. This procedural generation mechanism greatly enriches the training scenarios and solves the problem of traditional RL overfitting to a single environment.

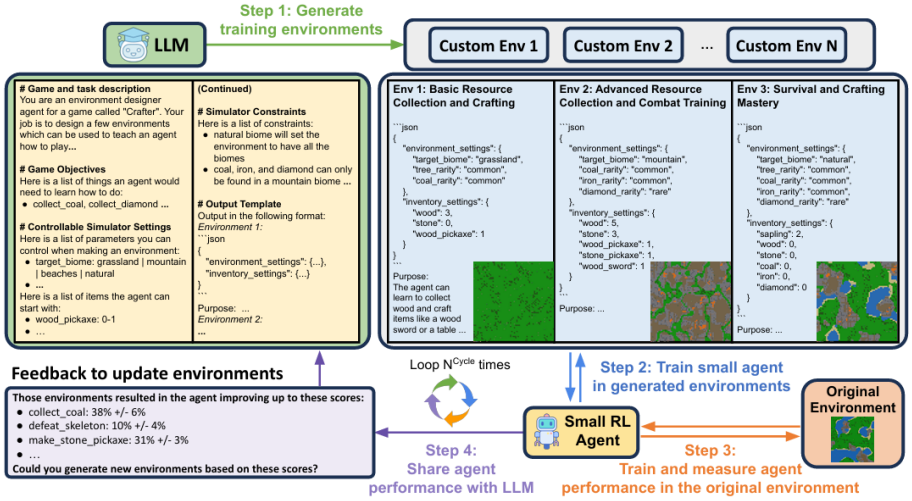

Fig. 2. Schematic diagram of the EnvGen framework [9].

2.3 Automation and density of reward engineering

The design of the reward function is known as the "black magic" of reinforcement learning. LLM effectively solves the problems of reward sparsity and alignment by leveraging its code generation and semantic understanding capabilities. At this stage, LLM primarily assumes the roles of "legislator" and "evaluator". Taking Auto MC-Reward [5] as an example (as shown in Figure 3), the system contains three LLM agents: Designers write the reward function code, critics check the code logic, and the analyzer reports errors based on runtime logs. This closed-loop mechanism allows the system to automatically iterate and generate high-quality, dense reward functions without human intervention. In addition, studies such as RLAIF vs. RLHF [7] have shown that using LLM to perform semantic scoring of agent behavior can achieve results comparable to expensive manual feedback.

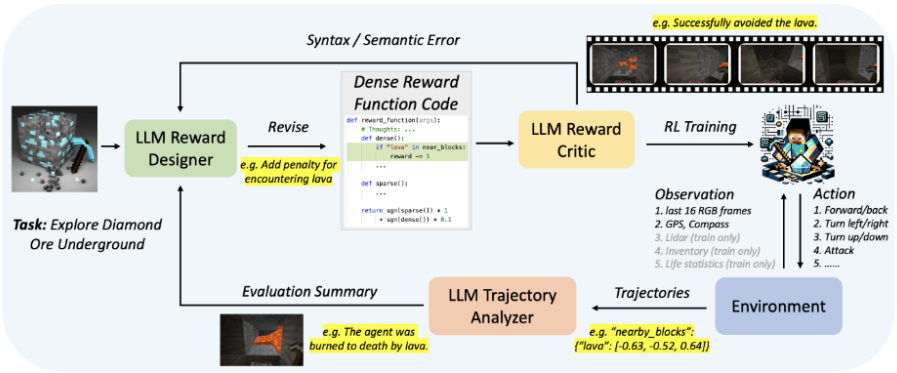

Fig. 3. Operational workflow of Auto MC-Reward[5]

2.4 Hierarchical reinforcement learning and hierarchical decision-making

In response to the problems of traditional RL lacking common sense guidance and having weak long-term planning capabilities, LLM can act as a "commander" to intervene in the decision-making level.

Hierarchical control and sub-goal decomposition: LLM can use its embedded world knowledge for macro-planning, decomposing complex tasks into a series of executable sub-goals. LLMAugmented Hierarchical Agents [10,11] proposed a two-layer architecture: the upper layer is a “commander” of an LLM that generates high-level instructions, and the lower layer is a “executor” of an RL agent that performs specific actions. Shek et al.'s Option Discovery Using LLM-guidedfurther applied this idea to the discovery of "Options" in hierarchical reinforcement learning (HRL)[12] . As shown in Figure 4, LLM is used to automatically identify and define reusable skill modules, which significantly improves the learning efficiency of complex tasks.

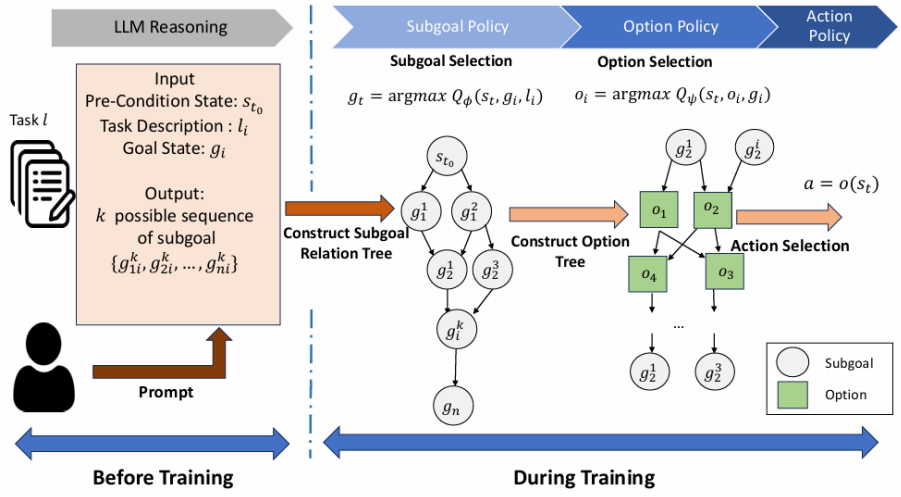

Fig. 4. LLM-guided hierarchical option discovery framework[12].

In high-risk fields such as autonomous driving, random exploration is unacceptable. As shown in Figure 5, LanguageMPC demonstrates how LLM acts as the decision-making brain, combined with Model Predictive Control (MPC), and uses common sense reasoning (such as "left turns must yield to straight-going vehicles") to directly guide vehicle behavior, giving the system strong interpretability [6]. Schultz et al.'s Mastering Board Games demonstrated that introducing prior value evaluation of LLM in Monte Carlo Tree Search (MCTS) can handle complex games like AlphaZero without training from Tabula Rasa[13].

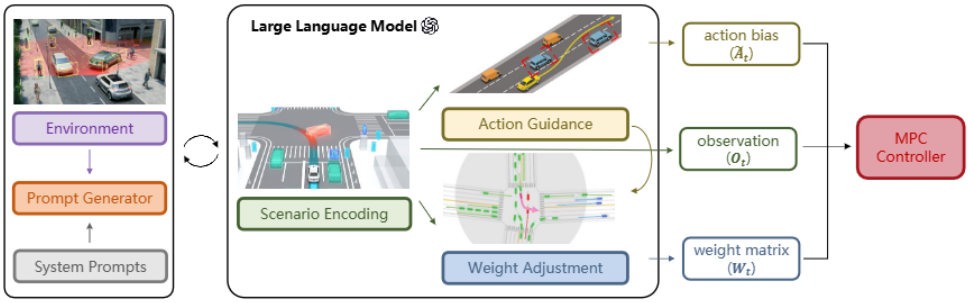

Fig. 5. Cognitive decision pathway of LanguageMPC [6].

2.5 Multi-agent collaboration and social evolution

Designing efficient communication protocols is extremely challenging in multi-agent systems. LLM-enabled agents use natural language as a universal interface, reconstructing the mechanism of group collaboration.

Based on the research framework of LGC-MARL (as shown in Figure 6), the LLM planner decomposes complex tasks into an action dependency graph and uses this as a basis to guide multiple agents to conduct structured collaboration and communication[14]. This enables intelligent agents to engage in complex negotiations and division of labor. In these frameworks, agents no longer exchange semantically meaningless vectors, but coordinate their actions through dialogue. Both evolved together through language interaction. This not only improves collaboration efficiency but also makes the behavior of the intelligent agent group highly interpretable. In addition, to cope with the uncertainty of dynamic environments, LLM has taken on the role of "dynamic intervention". The study of LLM-MEDIATED GUIDANCE shows that static programming alone is not enough. This study introduces a real-time supervision mechanism, allowing LLM to dynamically intervene through natural language during agent execution, effectively coordinating collaboration among multiple agent[15].

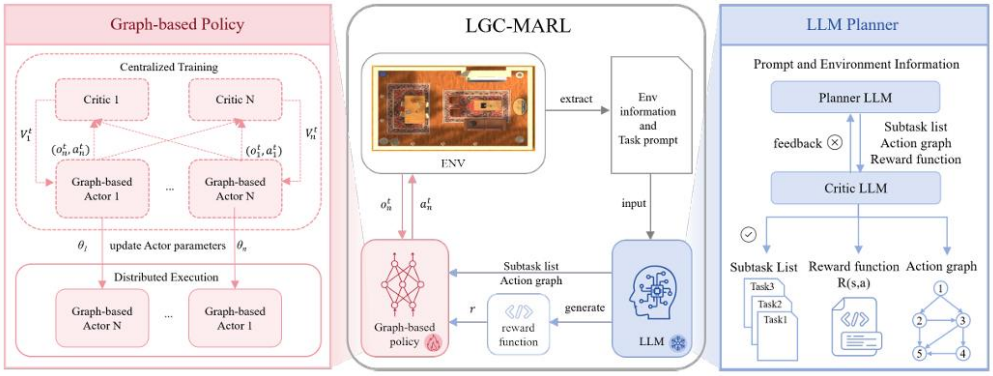

Fig. 6. LLM-based multi-agent collaboration framework[14]

Based on this, in order to improve the synergy of the group, as shown in Figure 7, the synergy mechanism revealed by Coevolving with the Other You evolves into group intelligence. In the dual role of "pioneer-observer", the intelligent agent realizes the leap from "following instructions" to "autonomous collaboration" through continuous language interaction and self-reflection and co-evolution by exchanging roles [16].

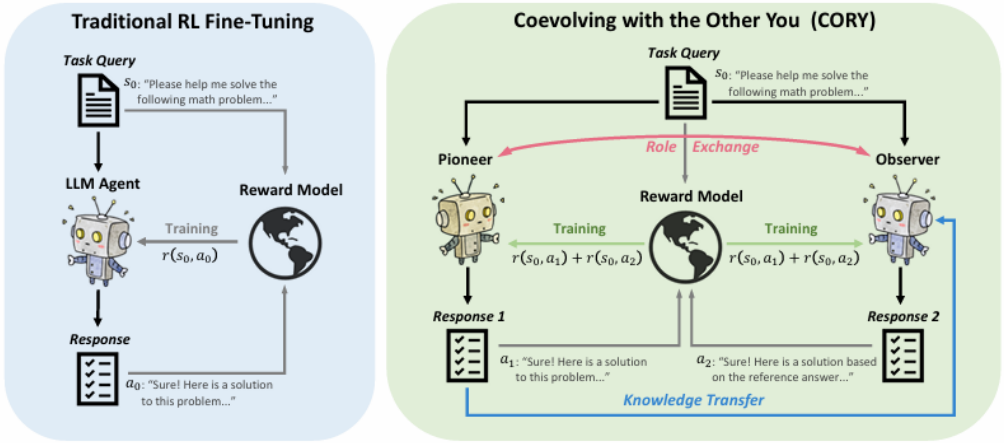

Fig. 7. Collaborative evolution framework of Coevolving with the Other You [16].

3 Challenges and Future Outlook

3.1 Challenges

Although LLM provides theoretical feasibility for reconstructing the entire lifecycle of reinforcement learning, it still faces challenges. The main issues are physical illusion and practicality. Since LLM is essentially a probability-based text generator rather than a physics simulator, it is easy to generate suggestions that violate common sense about physics (such as "retrieving objects from a distance"). Throughout the process, unverified outputs can lead to

incorrect environmental parameters or flawed reward, which in turn can mislead the agent into learning wrong behaviors. Secondly, there is a conflict between real-time performance and cost. Direct control often requires millisecond-level responses, while LLM inferences often take seconds and are very expensive. Directly calling a high-performance LLM at each step would result in unacceptable system latency and computational overhead, limiting its direct deployment in the underlying control layer. Finally, there are context windows and long-range memory. Complex RL tasks often involve long-term interactions (such as lifelong learning), generating massive amounts of data that exceed the carrying capacity of the fixed context window of LLM. In multi-agent collaboration or long-term planning, memory loss can lead to inconsistent agent strategies and an inability to maintain long-term goals.

3.2 Future Outlook

To address the challenges mentioned above, future "LLM-Enhanced RL" will be improved in the following four directions: To address the issues of real-time performance and cost, future solutions will no longer be limited to relying on small models to compensate for the shortcomings of large models, but will directly benefit from the rapid evolution of large models themselves. Just as GPU computing power and inference efficiency have exploded in the past few years, the inference cost of large models is decreasing by orders of magnitude, while the response speed is doubling. With the maturation of quantization technology, inference acceleration frameworks such as vLLM, and dedicated AI chips, future LLMs will have the ability to perform low-latency, low-power real-time inference at the edge. This will make it possible for LLM to directly intervene in the control of every frame of reinforcement learning as the underlying "general decision-making strategy," completely breaking the stereotype of "slow thinking".

The integration of native windows and external storage revolutionizes memory architecture. The agent will adopt a hybrid architecture of "native ultra-long context" and "external hierarchical memory". Drawing inspiration from computer storage architecture, a weight-independent "multi-level caching system" is constructed using a vector database. Similar to the "virtual context management" mechanism proposed by MemGPT, future systems will treat LLM as an operating system, dynamically managing an infinite stream of memory within a finite context window through paging technology [17]. Meanwhile, combined with the large-scale search enhancement technology demonstrated by RETRO, a vector database is used to store massive interaction history[18]. This architecture allows agents to retrieve key information from millions of steps ago on demand without increasing inference overhead, thus enabling true lifelong reinforcement learning(Figure 8).

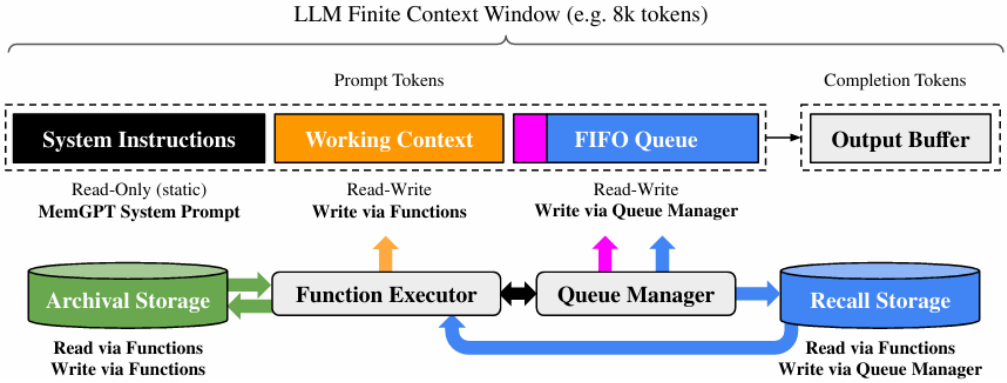

Fig. 8. Hierarchical memory management system of MemGPT: Towards LLMs as Operating Systems [17].

Towards a "world model". LLM will evolve from a “text generator” into a multimodal “world model” for understanding physical laws. The emergence of WALL-E 2.0[10] marks the beginning of this trend: This study introduced neural symbolic learning, successfully aligning the internal knowledge of the LLM with the physical dynamics of the environment. This will completely restructure the environment building phase, enabling agents to undergo low-cost offline training in a high-fidelity "mental world" generated by LLM, achieving zero-shot Sim-to-Real transfer, and ultimately solving the problem of physical implementation.

LLM-driven meta-reinforcement learning improves generalization ability. Traditional learning reinforcement learning (RL) often requires retraining from scratch when faced with new tasks, while meta-learning aims to "learn how to learn". With its vast amount of prior knowledge, LLM naturally possesses the potential to be a meta-learner. In the future, we will use the contextual learning capabilities of LLM so that when faced with a completely new task, LLM can quickly infer the task distribution and generate an adaptation strategy with only a few examples. This cross-domain transfer capability will significantly improve RL's few-sample generalization ability in open worlds.

4 Conclusion

Based on a "full life cycle" perspective, this paper constructs a taxonomic framework and systematically explores the substitution role of large language models for humans in four key aspects of reinforcement learning: environment construction, reward engineering, decision execution, and multi-agent collaboration. Research has found that at the basic construction layer, the code generation capability of large models enables the automated construction of simulation scenarios and dense rewards, effectively solving the problems of low sample efficiency and sparse rewards in traditional reinforcement learning; at the decision execution layer, the common sense reasoning capability of large models makes up for the shortcomings of poor generalization of traditional intelligent agents through the "hierarchical planning" paradigm. At the social evolution layer, the natural language interface reconstructs the collaboration mechanism among multiple agents, significantly improving the interpretability of the system. Future research should further focus on the integration of end-to-end world models and external

memory architectures to overcome the limitations of physical illusions and long-range memory. The deep integration of large language models and reinforcement learning achieves a closed loop between cognition and action, and will become a key path to AGI.

References

[1] Y. Cao et al., "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods," arXiv preprint arXiv:2404.00282, 2024.

[2] C. Sun, S. Huang, and D. Pompili, "LLM-based multi-agent reinforcement learning: Current and future directions," arXiv preprint arXiv:2405.11106, 2024.

[3] H. Pang, Z. Wang, and G. Li, "Large language model guided deep reinforcement learning for decision making in autonomous driving," arXiv preprint arXiv:2412.18511, 2024.

[4] M. Cao et al., "Beyond sparse rewards: Enhancing reinforcement learning with language model critique in text generation," arXiv preprint arXiv:2401.07382, 2024.

[5] H. Li et al., "Auto MC-Reward: Automated dense reward design with large language models for Minecraft," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2024, pp. 16426–16435.

[6] H. Sha et al., "LanguageMPC: Large language models as decision makers for autonomous driving," arXiv preprint arXiv:2310.03026, 2023.

[7] H. Lee et al., "RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback," arXiv preprint arXiv:2309.00267, 2023.

[8] X. Wang et al., "A survey on large language model based autonomous agents," Frontiers of Computer Science, vol. 18, no. 6, p. 186345, 2024.

[9] A. Zala et al., "EnvGen: Generating and adapting environments via LLMs for training embodied agents," arXiv preprint arXiv:2403.12014, 2024.

[10] S. Zhou et al., "WALL-E 2.0: World alignment by neurosymbolic learning improves world model-based LLM agents," arXiv preprint arXiv:2504.15785, 2025.

[11] B. Prakash et al., "LLM augmented hierarchical agents," arXiv preprint arXiv:2311.05596, 2023.

[12] C. L. Shek and P. Tokekar, "Option discovery using LLM-guided semantic hierarchical reinforcement learning," arXiv preprint arXiv:2503.19007, 2025.

[13] J. Schultz et al., "Mastering board games by external and internal planning with language models," arXiv preprint arXiv:2412.12119, 2024.

[14] Z. Jia et al., "Enhancing multi-agent systems via reinforcement learning with LLM-based planner and graph-based policy," arXiv preprint arXiv:2503.10049, 2025.

[15] P. D. Siedler and I. Gemp, "LLM-mediated guidance of MARL systems," arXiv preprint arXiv:2503.13553, 2025.

[16] H. Ma et al., "Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning," arXiv preprint arXiv:2410.06101, 2024.

[17] C. Packer et al., "MemGPT: Towards LLMs as operating systems," arXiv preprint arXiv:2310.08560, 2023.

[18] S. Borgeaud et al., "Improving language models by retrieving from trillions of tokens," in Proc. Int. Conf. Mach. Learn. (ICML), 2022, pp. 2206–2240.