ERFSL: An Efficient Reward Function Searcher via Large Language Models for Custom-Environment Multi-Objective Reinforcement Learning (Student Abstract)

Guanwen Xie[1*], Jingzehua Xu[1*], Yiyuan Yang[2], Yimian Ding[1], Shuai Zhang[3]

[1]MicroMasters Program in Statistics and Data Science, Massachusetts Institute of Technology, USA
[2]Department of Computer Science, University of Oxford, United Kingdom
[3]Department of Data Science, New Jersey Institute of Technology, USA
gwxie360@outlook.com, xjzh23@berkeley.edu, yiyuan.yang@cs.ox.ac.uk, yimiandingthu@gmail.com, sz457@njit.edu

Abstract

We propose ERFSL, an efficient reward function searcher using large language models (LLMs) for custom-environment, multi-objective reinforcement learning (RL). ERFSL generates reward components based on explicit user requirements and rectifies them, and iteratively optimizes the weights of these components based on textual context. Applied to an underwater data collection RL task, ERFSL corrects reward codes with only one feedback iteration per requirement, and acquires diverse reward functions within the Pareto set. ERFSL also presents robust capability for deviated weights and small-size LLMs such as GPT-4o mini. The full-text prompts, examples of LLM-generated answers, and source code are available at https://360zmem.github.io/LLMRsearcher/ .

Introduction

Reinforcement learning (RL) is useful for multi-objective tasks; however, designing complex reward functions remains challenging due to ambiguous and varied requirements(Hayes et al. 2022). Large language model (LLM)-aided reward function design has demonstrated remarkable performance in various scenarios, such as dexterous robots control (Ma et al. 2024; Zeng, Mu, and Shao 2024; Yu et al. 2023) and Minecraft playing (Wu et al. 2024; Li et al. 2024), yet issues such as incorrect code and imbalanced weights may arise with intricate tasks. To address these challenges, some approaches decompose complex tasks into several subtasks or skills, while providing clear task feedback accordingly (Mandi, Jain, and Song 2024; Triantafyllidis, Christianos, and Li 2024; Rho et al. 2024).

In this paper, we decompose the LLM-aided reward function generation into reward component design and weight assignment, employing LLMs as white-box searchers under textual context and clear feedback to fully leverage their semantic understanding capabilities. The ERFSL's architecture and key prompts are shown in Figure 1.

Methodology

Environment Description. Task description is a common part of most subsequent prompts, including text descriptions, environment code or APIs, and user requirements. We decompose user requirements into specific numerical objectives (e.g., achieving zero collisions for obstacle avoidance). To avoid ambiguous descriptions and facilitate human modification, we design a meta-prompt to allow LLMs to enhance the description quality.

Reward Code Generation. We borrow the LLM-aided reward code design frameworks, creating a reward component for each user requirement. However, the initial LLM-generated code may be incorrect due to the absence of prior knowledge about the environment and complex contexts. Therefore, we test each component separately and correct errors using the reward critic. LLMs can also fabricate necessary variables and prompt users to complete them, handling incomplete environment descriptions effectively.

Reward Weight Search. Multi-objective reinforcement learning requires a balanced scale of reward components. We first utilize a reward weight initializer to designate K=5 groups of weights to ensure the components' values are on the same scale by pre-calculating the values of reward components. After that, LLMs suggest K=5 weight adjustments and generate weight groups based on the training results summarized by the training log analyzer. To prevent ambiguity and potential redundancies in weight adjustments across multiple input groups, inspired by genetic algorithms, the searcher specifies the starting point and adjustment direction (increase/decrease/fine-tune) to mutate (adjust) each weight. For multiple weight adjustments, we conduct crossovers between the mutated input groups. Additionally, we separate the process of generating adjustment suggestions and output weight groups to shorten the prompt and ensure precise understanding and execution.

Experiments and Main Results

We utilize ERFSL to design reward functions in a zero-shot manner for a previous underwater data collection task via multiple-AUV (Zhang et al. 2024; Xu et al. 2024) trained by the TD3 RL algorithm, considering that marine tasks may present more challenges and possess more user requirements (Xie et al. 2024). We primarily use gpt-4o-2024-08-06 (denoted as **GPT-4o**), and also conduct tests with gpt-4o-mini-2024-07-18 (denoted as **GPT-4om**). Similar to Eureka (Ma et al. 2024), we design a baseline **EUREKA-M** that takes multiple full reward functions along with their training logs

*These authors contributed equally.

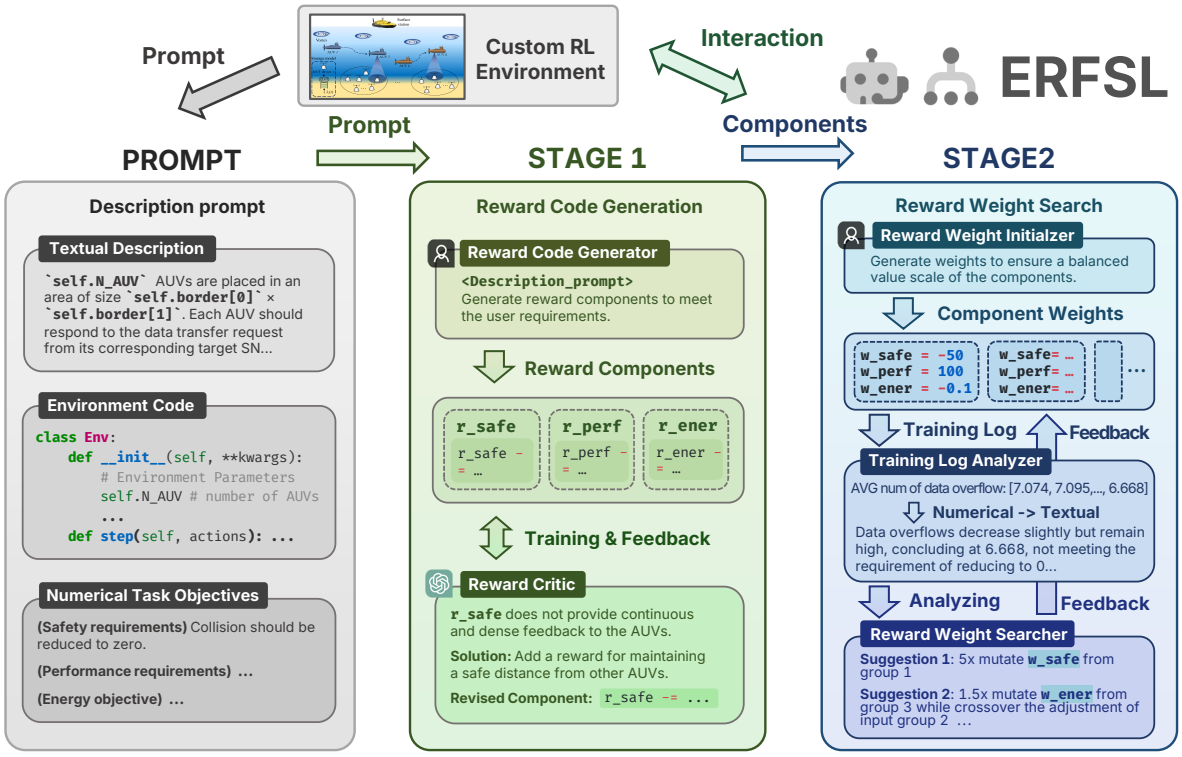

Figure 1: The main architecture and prompt examples of the proposed ERFSL.

Setting	GPT-4o	GPT-4o w/o TLA	GPT-4om
RWI		**0.40±0.49**	
RWI-UB	1.20±0.75	1.60±1.02	1.80±1.17
500x off	5.20±1.46	6.40±1.86	8.60±1.74

Table 1: The number of iterations of searching weights under different experiment settings, and each is performed 5 times.

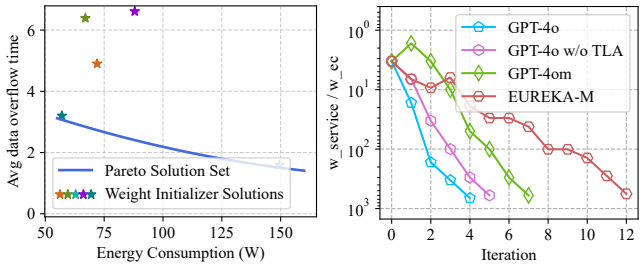

Figure 2: (a) Solutions generated from the reward weight initializer. (b) Change of maximum value of w_service/w_ec under different settings during iteration.

as input and outputs K=5 reward functions.

The reward critic can correct code errors with just one feedback per component, which avoids errors that are hard to correct when utilizing baselines like EUREKE-M, thanks to the explicit feedback coming with task splitting.

Reward weight initialization and search. Table 1 presents the number of searches required to meet user requirements. Figure 2(a) illustrates the initial weight groups generated by the reward weight initializer (denoted as **RWI**), while Figure 2(b) depicts the maximum value of w_service/w_ec in five weight groups during training, demonstrating the best performance among the five repetitions. Two of the five sets of solutions generated by the RWI achieve Pareto solutions. Even if the numerical reference and balance requirements (denoted as **RWI-UB**) are removed from the prompt of RWI, only 0-3 searches are required to meet user requirements. When the energy weight is increased by a factor of 500 (denoted as **500x off**), the reward weight searcher successfully identifies the problem and adopts a flexible step size strategy, requiring only an average of 5.2 adjustments. Moreover, EUREKA-M adopts small, random step sizes and increases weights in a highly random manner rather than decreasing the penalty for energy consumption, necessitating more searches.

The performance of GPT-4om. The splitting of the reward weight search process shortens the context, thereby avoiding the disadvantages faced by small-scale LLMs (Shen et al. 2024). Although the answer quality of the content generation prompt is slightly degraded and the reward weight initializer not functions, the performance of reward code generation and reward weight search is adequate. Although the step size selection lacks flexibility, the final performance still surpasses that of EUREKA-M.

Acknowledgments

Part of this work was done when Guanwen Xie and Jingzehua Xu were studying in the MicroMasters Program in Statistics and Data Science at Massachusetts Institute of Technology (MIT). We are very grateful to Yiyuan Yang and Shuai Zhang at University of Oxford and New Jersey Institute of Technology (NJIT) for their strong support, and to Miao Liu and Songtao Lu at IBM research and MIT-IBM Watson AI Lab and for their valuable advice, respectively. Additionally, we thank all anonymous reviewers for their constructive comments.

Ethical Statement

This paper presents work whose goal is to explore using LLMs to design reward functions for the multi-objective tasks. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Hayes, C. F.; Rădulescu, R.; Bargiacchi, E.; Källström, J.; Macfarlane, M.; Reymond, M.; Verstraeten, T.; Zintgraf, L. M.; Dazeley, R.; Heintz, F.; et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1): 26.

Li, H.; Yang, X.; Wang, Z.; Zhu, X.; Zhou, J.; Qiao, Y.; Wang, X.; Li, H.; Lu, L.; and Dai, J. 2024. Auto MC-Reward: Automated Dense Reward Design with Large Language Models for Minecraft. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ma, Y. J.; Liang, W.; Wang, G.; Huang, D.-A.; Bastani, O.; Jayaraman, D.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024. Eureka: Human-Level Reward Design via Coding Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Mandi, Z.; Jain, S.; and Song, S. 2024. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 286–299. IEEE.

Rho, S.; Smith, L.; Li, T.; Levine, S.; Peng, X. B.; and Ha, S. 2024. Language Guided Skill Discovery. *arXiv preprint arXiv:2406.06615*.

Shen, W.; Li, C.; Chen, H.; Yan, M.; Quan, X.; Chen, H.; Zhang, J.; and Huang, F. 2024. Small llms are weak tool learners: A multi-llm agent. *arXiv preprint arXiv:2401.07324*.

Triantafyllidis, E.; Christianos, F.; and Li, Z. 2024. Intrinsic language-guided exploration for complex long-horizon robotic manipulation tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 7493–7500. IEEE.

Wu, Y.; Min, S. Y.; Prabhumoye, S.; Bisk, Y.; Salakhutdinov, R. R.; Azaria, A.; Mitchell, T. M.; and Li, Y. 2024. Spring: Studying papers and reasoning to play games. *Advances in Neural Information Processing Systems*, 36.

Xie, G.; Wang, X.; Ding, Y.; Xu, J.; Ma, D.; Wang, J.; and Ren, Y. 2024. FISHER: An Efficient Sim2sim Training Framework Dedicated in Multi-AUV Target Tracking via Learning from Demonstrations. In *The 31st International Conference on Neural Information Processing*.

Xu, J.; Xie, G.; Wang, X.; Ding, Y.; and Zhang, S. 2024. USV-AUV Collaboration Framework for Underwater Tasks under Extreme Sea Conditions. *arXiv preprint arXiv:2409.02444*.

Yu, W.; Gileadi, N.; Fu, C.; Kirmani, S.; Lee, K.-H.; Arenas, M. G.; Chiang, H.-T. L.; Erez, T.; Hasenclever, L.; Humplik, J.; et al. 2023. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*.

Zeng, Y.; Mu, Y.; and Shao, L. 2024. Learning Reward for Robot Skills Using Large Language Models via Self-Alignment. *arXiv preprint arXiv:2405.07162*.

Zhang, Z.; Xu, J.; Xie, G.; Wang, J.; Han, Z.; and Ren, Y. 2024. Environment- and Energy-Aware AUV-Assisted Data Collection for the Internet of Underwater Things. *IEEE Internet of Things Journal*, 11(15): 26406–26418.