RainbowArena: A Multi-Agent Toolkit for Reinforcement Learning and Large Language Models in Competitive Tabletop Games

Extended Abstract

Yingzhuo Liu
Beijing University of Posts and Telecommunications
Beijing, China
liuyingzhuo86@bupt.edu.cn

Shuodi Liu
Beijing University of Posts and Telecommunications
Beijing, China
liushuodi@bupt.edu.cn

Hongsong Tang
Beijing University of Posts and Telecommunications
Beijing, China
ths@bupt.edu.cn

Yubing Ma
Beijing University of Posts and Telecommunications
Beijing, China
mayubing@bupt.edu.cn

Zikang Li
Beijing University of Posts and Telecommunications
Beijing, China
zikangli@bupt.edu.cn

Junge Zhang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
jgzhang@nlpr.ia.ac.cn

Liuyu Xiang
Beijing University of Posts and Telecommunications
Beijing, China
xiangly@bupt.edu.cn

Zhaofeng He
Beijing University of Posts and Telecommunications
Beijing, China
zhaofenghe@bupt.edu.cn

ABSTRACT

Tabletop games have gained little to no attention, despite offering a range of unique challenges compared to card or board games. We introduce RainbowArena, an open-source toolkit for reinforcement learning and large language models in competitive tabletop games. The goal of RainbowArena is to provide a unified, scalable platform that supports both Reinforcement Learning (RL) and Large Language Models (LLM), and push forward the research in tabletop games. RainbowArena consists of three modules: game, agent and evaluation. We design unified components and interfaces for various tabletop games. To better integrate with game environments, we devise an efficient self-play framework for RL agents, and a standardized prompt structure for LLM agents. Additionally, agents of all types can be evaluated within the evaluation framework. Finally, we evaluate various types of agents across different games and analyze the runtime efficiency for each game.

KEYWORDS

Reinforcement Learning; Large Language Model; Self Play; Tabletop Game

ACM Reference Format:
Yingzhuo Liu, Shuodi Liu, Hongsong Tang, Yubing Ma, Zikang Li, Junge Zhang, Liuyu Xiang, and Zhaofeng He. 2025. RainbowArena: A Multi-Agent Toolkit for Reinforcement Learning and Large Language Models in Competitive Tabletop Games: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Table 1: Comparisons between RainbowArena and others

	Features	RLCard	PyTAG	Openspiel	LLMArena	Ours
1	Multiplayer		✓	✓		✓
2	Lightweight & Customizable	✓				✓
3	RL agents	✓	✓	✓		✓
4	Self-play			✓		✓
5	LLM agents				✓	✓
6	Agent ranking			✓	✓	✓

Reinforcement learning (RL) [14, 18] and large language models (LLMs) [9, 16] have made remarkable progress across various domains, demonstrating their potential as powerful agents. However, tabletop games [5], despite offering a rich and complex environment for agents, have received relatively little attention in comparison to traditional board or card games such as Go [14] and Doudizhu [18].

By analyzing existing toolkits for board, card and tabletop games [1, 10, 17], we identify four key issues that need to be addressed: (1) Most research focuses on board or card games [14, 18], rather than tabletop games, mainly due to *the lack of toolkits offering a unified interface for such games.* (2) Recent success in tackling two-player games [15] has outpaced progress in n-player games [12]. One contributing factor to this issue is *the lack of a lightweight,*

easy-to-use, and customizable environment for multiplayer games. Although Openspiel [10] supports some multiplayer environments, it lacks the simplicity and user-friendliness that would facilitate broader use. Furthermore, certain game environments in Openspiel and PyTAG [1] are not implemented in Python, making customization challenging. (3) Game toolkits should facilitate the integration of RL agents, but training against solely fixed opponents limits their potential. Thus, self-play training is crucial to maximize agent performance. Additionally, LLM agents are capable of making decisions like RL agents, suggesting that game toolkits should support LLM agent integration as well. However, most existing toolkits either support only RL agents (e.g., RLCard [17], PyTAG, Openspiel) or only LLM agents (e.g., LLMArena [3]), with few toolkits supporting self-play training (e.g., Openspiel). Therefore, *developing a tabletop game toolkit that supports both RL and LLM agents, as well as self-play training*, is another critical challenge. (4) After training multiple agents, the next challenge is *how to evaluate their capabilities*. However, many toolkits only focus on evaluating head-to-head performance between two agents (e.g., RLCard, PyTAG).

To tackle the issues outlined above, we introduce RainbowArena, a toolkit for tabletop games that supports both RL and LLM agents. It provides a workflow from game construction to agent integration and learning, and then to agent evaluation. A comparison between our toolkit and other existing toolkits is presented in Table 1.

2 METHOD

RainbowArena consists of three main components: the game module, agent module, and evaluation module.

In the game module, we focus on tabletop games that are competitive, symmetric, adaptable to varying player numbers, and easy to model. Ultimately, six games with different levels of complexity and rules are selected: "Splendor", "Ticket to Ride", "Lama", "Gongzhu", "Wizard", and "Papayoo", most of which support multiplayer gameplay. We create a unified simulation environment and environment interface for all the selected games. Users can adjust game settings within the simulation environment submodule and customize observation representation, action encoding, and reward design in the environment interface submodule.

In the agent module, we implement two submodules: one for RL agents and the other for LLM agents. To enable self-play training for RL agents, a unified self-play framework is devised to support both two-player and multiplayer zero-sum games. This framework consists of three key components [2]: game simulation, meta-strategy solver, and best response. It is compatible with common self-play methods such as vanilla self-play [13], fictitious self-play [7], and policy space response oracles (PSRO) [11], with RL integrated as the best response solver. To accelerate the game simulation, an efficient meta-payoff matrix construction method is introduced, leveraging the properties of extensive-form games [7] and parallel computing frameworks. Users can extend this self-play framework with new self-play and RL methods to continually optimize agent performance. In the LLM agent submodule, we design a unified prompt structure comprising three components: system prompt, observation prompt, and action prompt, each tailored to the specific characteristics of the tabletop games. The structured approach allows users to easily integrate a variety of LLMs.

In the evaluation module, to mitigate the impact of randomness in tabletop games, we first construct a meta-payoff matrix based on the participating agents, representing the outcomes of different agent combinations. Based on the results from the meta-payoff matrix, three evaluation methods are implemented: Elo [6], TrueSkill [8], and Nash Clustering [4].

3 EXPERIMENTS

Table 2: The evaluation results of four types of agents

	RL w/. SP	RL w/o. SP	LLM	Random
Splendor	**17.28(1)**	10.71(2)	-3.44(4)	3.45(3)
Ticket to Ride	**17.76(1)**	4.64(3)	9.90(2)	-3.38(4)
Lama	**23.72(1)**	15.11(2)	11.88(3)	-1.61(4)
Gongzhu	**23.20(1)**	20.77(2)	11.06(3)	8.83(4)
Papayoo	**22.65(1)**	18.34(2)	14.98(3)	9.46(4)
Wizard	**28.23(1)**	11.29(3)	19.98(2)	-2.01(4)
Ranking(Avg.)	**1.00**	2.33	2.83	3.83

RainbowArena supports comparisons not only between the same type but also between different types of agents. As shown in Table 2, the RL agents enhanced with self-play training significantly outperform the other three agents across all games, highlighting that self-play training effectively enhances the capabilities of RL agents. Although LLM agents underperform RL agents in most games, they achieve comparable or even better results in some games (possibly due to the prior knowledge embedded in LLM agents). This indicates that although LLM agents have already excelled in many tasks, the complexity of tabletop games leaves significant room for further optimization of LLM agents.

Table 3: The runtime (seconds) of the baseline simulation method with our proposed accelerated method

	Baseline	Ours(×2)	Ours(×8)
Splendor	570.66	164.20	**59.26**
Ticket to Ride	56.30	19.45	**10.62**
Lama	130.99	28.18	**13.49**
Gongzhu	284.81	44.50	**20.16**
Papayoo	626.92	93.80	**36.73**
Wizard	130.99	28.18	**13.49**

The runtime of the baseline game simulation method and our proposed accelerated method (utilizing 2 or 8 processes) for the game simulation in each game is shown in Table 3. Each result in the table represents the average of three runs with different random seeds. As shown in the table, our method substantially enhances game simulation efficiency. Additionally, when sufficient computational resources are available, increasing the number of parallel processes further boosts performance.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2021ZD0110100), the National Natural Science Foundation of China (621760025, 62301066, U21B2045, 62206012), the Fundamental Research Funds for the Central Universities (2023RC72), and the Chinese Nutrition Society (CNS-YUM2024-120).

REFERENCES

[1] Martin Balla, George EM Long, Dominik Jeurissen, James Goodman, Raluca D Gaina, and Diego Perez-Liebana. 2023. Pytag: Challenges and opportunities for reinforcement learning in tabletop games. In *2023 IEEE Conference on Games (CoG)*. IEEE, 1–8.

[2] Ariyan Bighashdel, Yongzhao Wang, Stephen McAleer, Rahul Savani, and Frans A Oliehoek. 2024. Policy Space Response Oracles: A Survey. *arXiv preprint arXiv:2403.02227* (2024).

[3] Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Lijie Wen. 2024. Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments. *arXiv preprint arXiv:2402.16499* (2024).

[4] Wojciech M Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. 2020. Real world games look like spinning tops. *Advances in Neural Information Processing Systems* 33 (2020), 17443–17454.

[5] Raluca Gaina, Martin Balla, Alexander Dockhorn, and Raul Montoliu Colás. 2020. TAG: A tabletop games framework. CEUR Workshop Proceedings.

[6] Mark E Glickman and Albyn C Jones. 1999. Rating the chess rating system. *CHANCE-BERLIN THEN NEW YORK-* 12 (1999), 21–28.

[7] Johannes Heinrich, Marc Lanctot, and David Silver. 2015. Fictitious self-play in extensive-form games. In *International conference on machine learning*. PMLR, 805–813.

[8] Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: a Bayesian skill rating system. *Advances in neural information processing systems* 19 (2006).

[9] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[10] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. 2019. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453* (2019).

[11] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems* 30 (2017).

[12] Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. 2021. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *International Conference on Machine Learning*. PMLR, 7480–7491.

[13] Arthur L Samuel. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development* 3, 3 (1959), 210–229.

[14] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[15] 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.

[16] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data* 18, 6 (2024), 1–32.

[17] Daochen Zha, Kwei-Herng Lai, Yuanpu Cao, Songyi Huang, Ruzhe Wei, Junyu Guo, and Xia Hu. 2019. Rlcard: A toolkit for reinforcement learning in card games. *arXiv preprint arXiv:1910.04376* (2019).

[18] Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. 2021. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *international conference on machine learning*. PMLR, 12333–12344.