AutoDiner: Empowering Restaurant Simulations with Advanced LLMs for Enhanced Agent Collaboration

Anonymous ACL submission

Abstract

As large language models (LLMs) continue to demonstrate impressive reasoning capabilities, LLM- based multi-agent has become an increasingly compelling area of research. Despite the potential, the field faces a notable gap: the scarcity of LLM-based simulators tailored for realistic, multi-agent interactions. Most existing multi-agent simulators are missing textual interfaces and quantitative evaluation metrics, limiting the assessment of complex interactions between agents. Our proposed simulator, `AutoDiner`, replicates a detailed restaurant management scenario requiring advanced communication and teamwork among agents, providing a uniquely realistic and complex research environment. `AutoDiner` not only fosters intricate agent interactions but also incorporates varying levels of difficulty and performance metrics for comprehensive benchmarking. These features make `AutoDiner` an exemplary platform for advancing the understanding and capabilities of LLM-based agents in navigating complex tasks and enhancing cooperative strategies in realistic settings.

1 Introduction

Large language models (LLMs) have shown advanced reasoning and instruction-following skills, spurring more research into LLM-driven autonomous agents (Brown et al., 2020; OpenAI, 2023; Wu et al., 2024). A key strength of LLM-based agents is their ability to communicate and collaborate, solving complex tasks, and forming communities (Park et al., 2023a; Qian et al., 2023).

Most existing virtual multi-agent environments (Bansal et al., 2017; Savva et al., 2019; Carroll et al., 2020; Gong et al., 2023) are tailored for conventional reinforcement learning methods and do not feature text-based interfaces. Furthermore, the complexity of these environments (Wang et al., 2023; Zhu et al., 2023) is limited, with communication between agents not being essential for the completion of tasks, and doesn't reflect the real-world need for agents with many different roles to collaborate. Several work has explored along this direction. Park et al., 2023b developed an LLM-based simulator to study LLM social behaviors and collaboration roles, whereas Qian et al., 2023 examined software engineering contexts. However, these environments fall short of a overall quantitative assessment of multi-agent capabilities.

To fill the gap, we introduce `AutoDiner`, a simulator tailored for LLM multi-agent research. Unlike earlier kitchen-focused environments from Carroll et al. (2020); Gong et al. (2023), `AutoDiner` simulates a comprehensive restaurant management scenario where agents work together handling customer service from arrival to departure. Agents must act as both players and NPCs, adapting strategies for planning and coordination, not just to maximize profits but also to satisfy customer demands. This more intricate and realistic setup evaluates LLM decision-making and simulates human-like teamwork.

Through our evaluation, we found that even strong LLMs like GPT-3.5-Turbo struggle with task repetition and lack effective collaboration skills, while GPT-4, despite better planning, struggles with resource management. Our contributions are:

- We develop an innovative simulator `AutoDiner`, which is designed for LLM-based agents to manage a restaurant. This environment necessitates intricate communication and collaboration among mult-agents, spurring multi-agent research.
- We propose a unified multi-agent framework that allows models selection and prompt customization. This allows for a fair comparison across different LLMs and agent algorithms.
- Our environment offers a range of settings and tasks with different levels of difficulty, serving as a versatile platform for evaluating LLM-based agents.

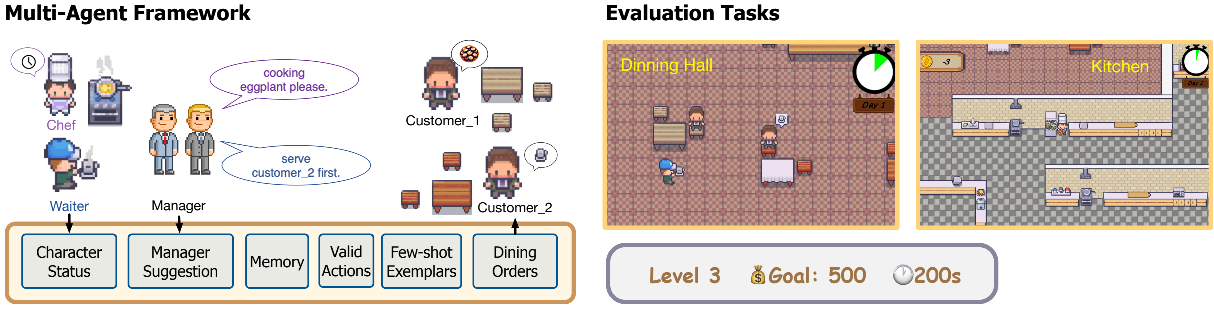

Figure 1: Overview of AutoDiner. **Left:** Illustration of the proposed multi-agent framework, which encompasses different characters. Memory of previous interactions, valid actions and few-shot exemplars are also provided to agents. **Right:** Illustration of game scene, where each level requires players to maximize profit within limited time.

2 AutoDiner

In `AutoDiner`, we present a simulation that leverages large language models to enable agents to collaboratively manage a virtual restaurant, aiming to maximize profitability through improved customer satisfaction and reduced costs. This section is organized into four parts: §2.1 introduces the agent roles within the simulation; §2.2 describes the environment; §2.3 outlines the task settings; and §2.4 discusses varying difficulty levels designed for agent evaluation.

2.1 Agents

`AutoDiner` consists of four distinct types of agents: the chef, the waiter, the customer, and the manager.

- **Chef.** The chef is tasked with the job of preparing dishes. He must use appropriate ingredients and kitchen tools to make meals while producing a large amount of food in a restricted time frame, controlling expenses, and reducing waste.
- **Waiter.** The waiter must actively communicate with customers to complete orders and serve food in a timely manner. The effectiveness of the waiter is directly related to customer satisfaction and the smooth running of the restaurant.
- **Customers.** Customers are each equipped with their own needs, simulating complex and varied accommodations in reality.
- **Manager.** The manager oversees the entire restaurant. He is responsible for determining the priority in attending to customers and making timely adjustments to staff behavior. This role involves strategic decision making to ensure restaurant efficiency and profitability.

2.2 Environments

The 2D restaurant environment in our simulation is crafted using Unity, offering a highly interactive and realistic setting split into two main areas: the kitchen and the dining hall. The kitchen is equipped with a variety of essential kitchen appliances and utensils for meal preparation, while the dining hall features tables and chairs arranged to accommodate the continuous influx of customers seeking to dine. Agents within this environment must work effectively to perform hosting tasks, ensuring that customers are promptly seated, served, and satisfied with their dining experience. This setup not only challenges agents to work together efficiently, but also provides a dynamic backdrop for testing and improving their decision-making and service skills in a simulated real-world scenario.

Figure 1 shows the simulator operating in a flexible multi-agent framework. This framework encompasses detailed character settings and statuses that inform agents of their abilities and constraints, as well as item properties that are specific to their roles, for instance, chefs are aware of the state of kitchen tools, while waiters are keyed into the dining hall's seating situation.

The range of actions available to agents includes basic movements as well as more complex interactions such as picking up plates, opening the oven or communicating with customers. In addition, the framework includes a memory system that allows agents to draw on past experience to inform current decisions, and we incorporate in-context learning for agents, providing contextually relevant examples within their prompts to guide them towards more accurate and standardised action generation.

Central to the framework is the Manager Suggestion component, which equips the virtual manager with the ability to guide and prioritize the actions of the chef and waiter, similar to a human player directing a team.

Moreover, we establish a streamlined programming interface, simplifying the process of customizing prompts and models. This interface is designed to be user-friendly and intuitive, enabling users to effortlessly tailor the simulation to their specific research or training needs. To further augment the usability and accessibility of the environment, a comprehensive User Interface (UI) has also been developed. Through this interface, users can not only modify agent behaviors and environmental parameters but also visually monitor the effects of these changes in real-time.

2.3 Task Definition

In our simulated environment, there are n agents, where the size of n varies with the number of customers. At each time step, all n agents take an action a_i based on their current state s_i and observation o_i. These actions can include moving, interacting, or communicating. Chef, waiter, and the manager need to collaborate to enhance the service provided to each customer, thereby increasing the reward r_i obtained from each customer. This, in turn, aims to improve the overall objective: the total revenue G of the restaurant.

Specifically, the restaurant's profit G is controlled by the income from serving customers, $\sum_i r_i$, and the restaurant's costs C, such that $G = \sum_i r_i - C$. If customer i initiates j orders, with each order having a maximum waiting time of t_j and a base income of p_j, and the income from each order is related to the time τ_j taken from the order being placed to being served, denoted by the function $f(\tau)$, then $r_i = \sum_j f(\tau_j)$. The function $f(\tau_j)$ is defined as, where α and γ are constants.

$$f(\tau_j) = \begin{cases} (1+\alpha)p_j & \text{if } \tau_j \leq \gamma t_j \\ p_j & \text{if } \gamma t_j < \tau_j \leq t_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

2.4 Difficulty Level Setup

To create a comprehensive evaluation framework, the simulation was structured into levels of progressively increasing difficulty. Each level was designed with distinct parameters: allotted time for task completion, type of food, and number of customers.

Additionally, to further challenge the agents and enhance the simulation's realism, each level introduces a unique special condition. These special conditions are cumulative, meaning that with each advancement to a higher level, the agents are not only faced with the new conditions introduced at that stage but must also navigate the complexities carried over from all previous levels. The detailed configurations of these levels are documented in Table 1. This structure not only challenges the agents but also provides a dynamic framework for assessing their problem-solving abilities, strategic planning, and collaborative efficiency in a controlled yet varied setting.

3 Experiments

We use GPT-3.5-Turbo and GPT-4 to empower agents in our environment. We set the temperature to zero to facilitate the reproduction of our results. During each round of action, each agent receives a scenario-specific prompt depending on their respective roles.

3.1 Metrics

We evaluate agents' performance within our environment using three metrics: Revenue, Order Completion Rate, and Bonus.

- **Revenue (Rev).** Revenue represents the amount of money earned during operation. Revenue is awarded for fulfilling specific customer orders and is deducted to account for the cost of ingredients used in the preparation of food, as calculated in Equation 1. We assigns a selling price for three types of dishes at 10, 20, and 30 units respectively, with corresponding ingredient costs at 3, 5, and 8 units. Additionally, beverages in the simulation are priced at 5 units, providing a simpler, yet integral component to the overall revenue calculation.
- **Order Completion Rate (OCR).** The Order Completion Rate metric quantifies the proportion of customer orders that are completed satisfactorily versus the total orders received.
- **Bonus.** Bonuses are awarded for exceptional performance. As detailed in the description of the function $f(\tau)$ in Section 2.3, if an order is completed within a time frame of γt_j, a bonus of αp_j is awarded. Here we set $\alpha = 0.2, \gamma = 0.5, t_j = 120$ seconds.

Both Revenue and Order Completion Rate measure the efficiency of agent collaboration, and the Bonus metric further quantifies models' ability to go above and beyond in optimizing restaurant operations and enhancing customer experience.

Table 1: Comprehensive Breakdown of Simulation Levels.

	Time (seconds)	Dish Types	Number of Customers	Special Conditions
Level 1	120	1	7	None
Level 2	180	2	10	Customers may order drinks before meals
Level 3	210	3	15	Customers may seek recommendations before ordering
Level 4	270	3	-	Continuous flow of customers

3.2 Results

The experimental results for GPT-3.5-Turbo and GPT-4 at varying levels are presented in Table 2. A detailed analysis of the agent trajectories reveals that GPT-3.5-Turbo often produces redundant directions and shows weak management capabilities, resulting in the failure to successfully complete the collaborative tasks of cooking and serving. This led to a consistent score of zero at all levels for GPT-3.5-Turbo.

In comparison, GPT-4 demonstrates superior planning capabilities. However, it struggles with certain real-life scenarios, such as remembering to use a plate to serve the cooked dishes. Also, GPT-4 often shows performance degeneration when performing repetitive tasks, such as suceeding to make a beverage in first round of service yet struggles in the second round. Another issue observed is that GPT-4 has difficulty efficiently managing resources. For example, when the customers are still waiting for meal, the chef starts to wash dishes, wasting both time and resources. Despite the advanced abilities of GPT-4, performing real-world task through collaboration proves challenging.

Our evaluation results show that our simulated environments are challenging even for SOTA LLMs and emphasizes the importance of evaluating LLM-based multi-agents in realistic settings.

Table 2: Experimental Results Comparison Between GPT-3.5-Turbo and GPT-4. OCR stands for the order completion rates and Rev. denotes revenue metrics. †The dash "-" for Level 4 is due to a continuous flow of customers, thus omitting maximum metrics.

	GPT-3.5-Turbo			GPT-4			Max
	Rev.	OCR	Bonus	Rev.	OCR	Bonus	Rev.
Level 1	0	0 / 7	0	13	2 / 7	2	63
Level 2	0	0 / 10	0	26	3 / 10	3	250
Level 3	0	0 / 15	0	37	2 / 15	4	510
Level 4	0	0 / -	0	43	3 / - †	4	-

4 Related Work

LLM-Based Multi-Agent. Multi-agent interactions typically fall into two paradigms: either collaborative, where agents share a common goal, or competitive (Guo et al., 2024). Both the paradigms call for the strong communication and planning ability of LLMs through language (Lazaridou et al., 2016; Havrylov and Titov, 2017). In our work, we focus on the cooperative abilities of agents. Previous work has explored along this direction, covering diverse tasks including cooperative development of softwares (Qian et al., 2023), improving reasoning with debate (Du et al., 2023; Tang et al., 2023; Sun et al., 2023), and playing diplomacy games (Mukobi et al., 2023; , FAIR), featuring the strength of collaborative communication skill of LLM-based agents. Our work explores the multi-agent collaboration through various demanding tasks towards a common goal.

Multi-Agent Simulators. Despite of extensive research in single-agent simulators and benchmarks (Côté et al., 2019; Fan et al., 2022; Yao et al., 2022; Liu et al., 2023; Ma et al., 2024), the evaluation of LLM-based multi-agent interactions is a relatively underexplored. Reinforcement learning community proposes several multi-agent challenges (Rashid et al., 2020; Cordasco et al., 2018), yet they often lack a textual interface among agents. Recently, Park et al. (2023b); Zhang et al. (2023); Chen et al. (2023); Li et al. (2023) adapt LLMs as NPCs and propose SandBox-like simulators. These simulators often fall short of offering a quantitative measurement of agent behaviors or a clearly defined common goal. We aim to address these issues by providing multi-level tasks as well as a unified multi-agent framework for multi-agent evaluation.

5 Conclusion

In conclusion, we introduced `AutoDiner`, a novel simulation environment designed to advance research in LLM-based multi-agent systems. `AutoDiner` sets the stage for future studies to explore more complex interactions, develop more sophisticated strategies for agent collaboration, and further refine the models for enhanced performance. we anticipate that environments like `AutoDiner` will push forward the boundaries of agent research.

6 Limitation

One notable limitation of our current simulation environment is its high operational cost. Utilizing advanced LLMs to empower agents involves significant computational resources, which can lead to increased expenses for research and development. This cost factor may limit accessibility for smaller research teams or individuals without the requisite funding or computational infrastructure.

Additionally, while the simulation provides a robust platform for exploring agent collaboration and decision-making in a restaurant management context, the range of business management elements integrated into the simulation is not exhaustive. Certain aspects of restaurant operations, such as financial management, staff training, and customer relationship management, are not fully developed.

References

Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. 2017. Emergent complexity via multi-agent competition. *arXiv preprint arXiv:1710.03748*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. 2020. On the utility of learning about humans for human-ai coordination.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*.

Gennaro Cordasco, Vittorio Scarano, and Carmine Spagnuolo. 2018. Distributed mason: A scalable distributed multi-agent simulation environment. 89:15–34.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2019. Textworld: A learning environment for text-based games. In *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*, pages 41–75. Springer.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. 2023. Mindagent: Emergent gaming interaction.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*.

Nian Li, Chen Gao, Yong Li, and Qingmin Liao. 2023. Large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *arXiv preprint arXiv:2401.13178*.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. 2023. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*.

OpenAI. 2023. Gpt-4 technical report.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023a. Generative agents: Interactive simulacra of human behavior.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023b. Generative agents: Interactive simulacra of human behavior. abs/2304.03442:1–22.

Chen Qian, Xin Cong, Wei Liu, Cheng Yang, Weize Chen, Yusheng Su, Yufan Dang, Jiahao Li, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. 21:4292–4301.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. 2019. Habitat: A platform for embodied ai research.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. Corex: Pushing the boundaries of complex reasoning through multi-model collaboration. *arXiv preprint arXiv:2310.00280*.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2023. Medagents: Large language models as collaborators for zero-shot medical reasoning. *arXiv preprint arXiv:2311.10537*.

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.

Zhiyong Wu, Chengcheng Han, Zichen Ding, Zhenmin Weng, Zhoumianze Liu, Shunyu Yao, Tao Yu, and Lingpeng Kong. 2024. Os-copilot: Towards generalist computer agents with self-improvement. *arXiv preprint arXiv:2402.07456*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. 2023. Proagent: Building proactive cooperative agents with large language models. abs/2308.11339.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. 2023. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory.