Economics Arena for Large Language Models

Shangmin Guo[*] [1]
s.guo@ed.ac.uk

Haochuan Wang[*] [2]
2021211239@stu.hit.edu.cn

Haoran Bu[*] [3]
buhaoran2002@bupt.edu.cn

Yi Ren[4] **Dianbo Sui**[2] **Yuming Shang**[3] **Siting Lu**[1]

Abstract

Recent economics literature suggest that large language models (LLMs) are capable of playing various types of economics games. Following these works, we propose to explore competitive games as an evaluation for LLMs to incorporate multiplayers and dynamicise the environment. In our experiments, we find that most of LLMs are rational in their strategies that can increase their payoffs, but not as rational as indicated by Nash Equilibria (NEs). Moreover, when game history are available, certain types of LLMs, such as GPT4, can converge faster to the NE strategies, which suggests higher rationality level in comparison to other models. In the meantime, certain types of LLMs can win more often when game history are available, and we argue that the winning rate reflects the reasoning ability with respect to the strategies of other players. In this work, we provide an economics arena for the LLMs research community as a dynamic simulation to test the rationality and the strategic reasoning abilities of LLMs.

1 Introduction

Recently, economists, such as Horton [8], Phelps & Russell [16], Akata et al. [1], have applied LLMs to many typical economics games, either cooperative or competitive. Through studying their behaviours in the ultimatum game and the prisoner's dilemma games, Fu et al. [7] has also showcased LLMs' emergent price bargaining ability. Inspired by their works, we propose to evaluate LLMs in simulated number-based competitive games.

In our work, we build a benchmark named `EconArena`, using competitive games (e.g. beauty contests, private-value second price auctions), which can be expanded in a broader project. These games provide a simple baseline, allowing for an environment where multiple LLMs can interact with one another, and they have clearly defined NEs that comprise of actions in numeral format, thereby generating quantitative performance metrics that can be easily used to "score" the models.

We use rationality as the primary determinant of performance to verify whether the models can perform rationally. Here, we follow the most common rationality assumption used in economics, where a rational agent should aim to maximise one's utility [11]. Also, we care about whether LLMs can, like humans, win games through learning and reasoning about the strategies of other players [17], thus proposing to measure the strategic reasoning ability of LLMs by exposing them with game histories. Furthermore, we implement additional experiments with our `EconArena`, such as dynamicise the game configurations, to verify their capability to dynamic environments.

We contribute to the existing literature in a few ways: Current studies focused on self-play of certain types of LLM, but we open up the possibility to evaluate and compare the respective performance

[*] Equal contribution. [1] University of Edinburgh. [2] Harbin Institute of Technology. [3] Beijing University of Posts and Telecommunications. [4] University of British Columbia.

of different LLMs. Besides, while existing works draw a parallel between the game results and human behaviours, there are no standard performance indicator that are more solidly and theoretically grounded that can be used to assess the performance of the LLMs. Our `EconArena` exactly provides a set of quantitative metrics for measuring the performance of LLMs.

2 Economics Arena

2.1 Single-round Competitive Games with In/complete Information

At the moment, we consider only single-round games in `EconArena` , such that the LLMs can be used in an "off-the-shelf" fashion. In this way, we introduce the least amount of prompt engineering onto the performance of LLMs, and put all relevant game information in a single prompt to avoid potential influence from techniques like dialogue management [5]. When running LLMs in `EconArena`, every single run of the games is only a single-round dialogue, and no multi-round dialogue is needed. Considering the game, we focus on those with unique pure NE in the initial version of our economics arena. Such games allow conflicts between agents, and the winning strategy is not necessarily a strategy that brings positive payoffs. Having a unique NE ensures there exists an optimal strategy agents can adopt, and no one can obtain better outcome by deviating. To be more specific, we implemented the beauty contest games, and the second-price auctions, in the alpha version of `EconArena`[1]. More formal descriptions of the games are provided in Appendix A.1.

2.2 Metrics of Interest

We hereby propose and list some metrics arisen in our economics arena that are interesting for either LLM researchers or economics researchers. Discussion about the following metrics in further details can be found in Appendix A.3.

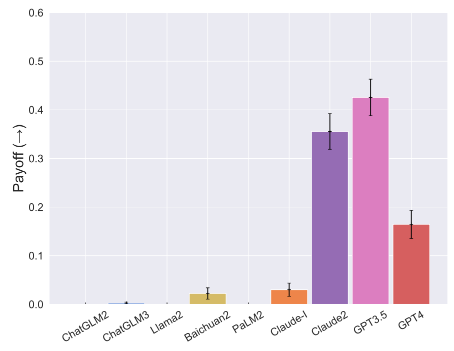
- **Payoff changes over games**: The first and foremost metric of various research interest is the change of payoffs over the arena games, which is measured by u_i^g, i.e. the utility function from game g to player i. Furthermore, since all arena games have only one unique NE, suppose the utility of player i under game g is $\bar{u}_i^g$, $u_i^g - \bar{u}_i^g$ can also be of interest as it can reflect how close the player's strategy s_i^g is close to the NE which is the optimal strategy when all players are rational.

- **Strategies over game configurations**: Strategies of the player i in a game g with varying configurations c, $s_i^g(c_j)$ where j indexes different configurations, is also an interesting metric, as it can reflect whether $s_i^g(c_j)$ is *adaptive* or *consistent* over game configurations c_j where $j \in \mathcal{Z}^+$. Also it is interesting to observe whether a player i plays different strategies when the opponents changed. When all other players in a game are rational, a more rational strategy should always be preferred, as irrational strategies definitely lead to lower payoffs. On the other hand, when other players in a game are mostly irrational, a rational strategy may not be a winning strategy.

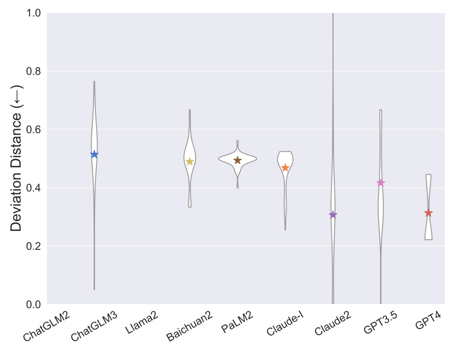
- **Strategies over game history availability**: Another importance factor for the performance of the LLM-based agents is the in-context learning capability [22]. To test whether and how well the players can improve their strategies through in-context learning, our economics arena can provide the game history for the previous runs in the same session.

3 Experiment Results

In our experiments using `EconArena`, we take the following LLMs as players: 1) `GPT4` from [15]; 2) `GPT3.5` from [14]; 3) `Claude2` from [4]; 4) `Claude-I` from [3]; 5) `PaLM2` from [2]; 6) `Llama2` from [19]; 7) `Baichuan2` from [23]; 8) `ChatGLM2` from [6, 25]; 9) `ChatGLM3` from also [6, 25]. Given the uncertainty nature of LLMs' responses, we run multiple times for all experiments. Due to the page limits, details about the prompts we use and more experimental results are provided in Appendix B and Appendix C.

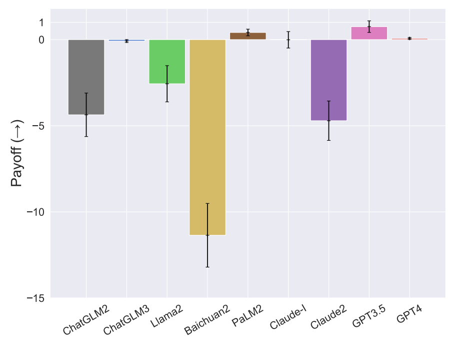


(a) Melee environment

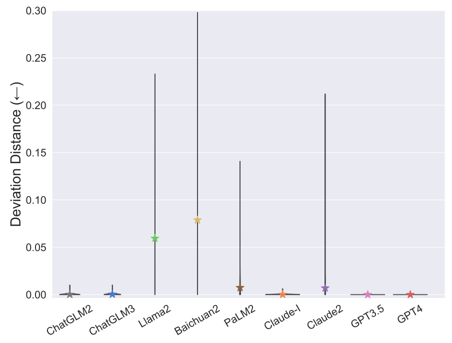


(b) Rational environment

Figure 1: The performance of LLMs in beauty contest games playing against different types of opponents. "Melee environment" corresponds to the case where the LLMs are playing against each other, while in the "Rational environment", 1 certain LLM plays against 4 hard-coded rational agents.



(a) Melee environment



(b) Rational environment

Figure 2: The performance of LLMs in second price auction games playing against different types of opponents. Note that Figure 2b is a vilolin graph, same as Figure 1b.

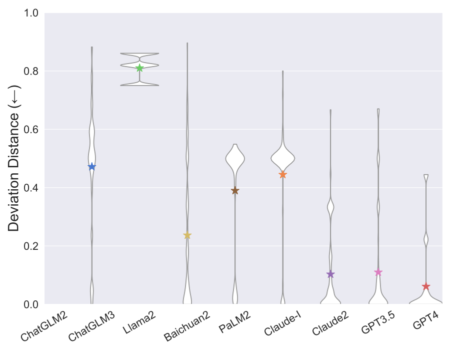
3.1 Non-maximally Rational Behaviours of LLMs in Economics Arena

To check the rationality of LLMs, we investigate whether the LLMs in `EconArena` are always seeking to increase their payoffs. Results of LLMs playing beauty contests and second-price auctions are shown in Figure 1 and 2 respectively. The results are from 150 independent sessions.

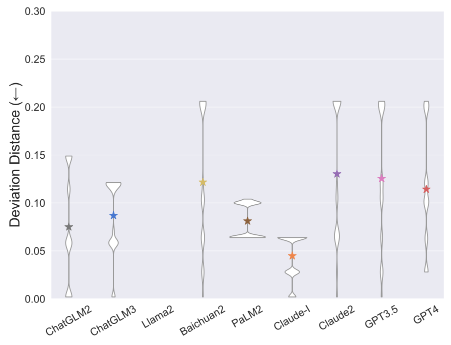
As shown in Figure 1a, the mean payoffs are almost 0 in the beauty contests for `ChatGLM2`, `ChatGLM3`, `Llama2` and `PaLM2`, but positive for the rest, and `GPT3.5` outperforms all. However, this comparison does not imply stronger rationality of `GPT3.5` among the LLMs straight away, as there are two potential causes behind the agents' behaviours: 1) the LLMs themselves are non-maximally rational; 2) the LLMs believe that the opponents they are facing are not rational, thus they choose non-maximally rational strategies. To disentangle the reasoning behind this, we explore another set-up: each LLM is playing against 4 hard-coded rational agents, and they were each given explicit prompts that they are playing with rational opponents. Henceforth, this is denoted as the "rational environment", and serves as the baseline for all comparisons. Figure 1b displays that despite knowledge of rational opponents, all models still deviate away from NE, implying that they are not maximally rational.

For second price auctions in Figure 2a, `GPT3.5` receives the highest average payoffs and `Baichuan2` obtains the lowest. Similarly, we cannot judge the rationality of the models solely based on this result. In Figure 2b, we again illustrate a "rational environment" and investigate the

[1]We will introduce more types of games in the updates version of `EconArena`, not only competitive games, but also cooperative games.

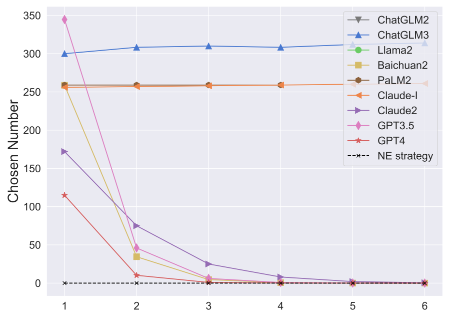


(a) Beauty contest games

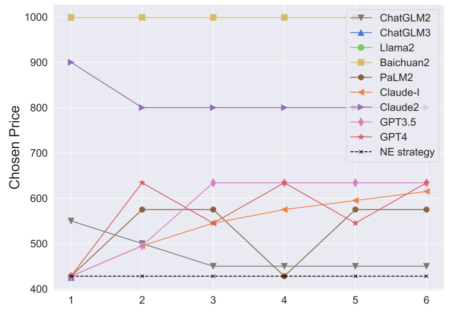


(b) Second price auctions

Figure 3: Deviation distance (↓) from NEs in the “Rational environment" with history. In these experiments, we reveal a maximum 3 runs of history to the LLMs.



(a) Beauty contest games



(b) Second price auctions

Figure 4: The path of chosen actions over the 6 runs within a session when a maximum 3 runs history is given to the LLMs, where the x-axis represents the run's index.

actual payoffs against NE payoffs via deviation distance. Most models can obtain payoffs very close to NE, except for `Llama2` and `Baichuan2`. `PaLM2` and `Claude-I`'s deviation distances differ from 0 only slightly. By controlling for the rationality perception about other players, it is possible to infer that in the second-price auction, `ChatGLM2`, `ChatGLM3`, `Claude-I`, `GPT3.5` and `GPT4` are behaving rationally on average.

3.2 Strategic Reasoning through Game History in Economics Arena

To show strategic reasoning ability of LLMs, we reveal game history to the agents. The “rational environment" serves a baseline. Through learning about other agents' *past* behaviours, it is expected for LLMs to reason about their strategies in view of the perceived rationality of their opponents. Altogether, we conducted 6 runs per session, and a maximum 3 runs of history were revealed.

Comparing the case with history (Figure 3a) to without (Figure 1b) for beauty contests, almost all LLMs, apart from `ChatGLM3` and `Claude-I`, show decrease in mean deviation. `GPT4` has the lowest mean deviation. We can interpret that all LLMs learnt to behave closer to NE. Figure 4a shows there is convergence in action with inclusion of history. The convergence speed is substantially faster for `GPT3.5` and `Baichuan2` among all the models. The slower convergence speed of `GPT4` can be interpreted as it being sufficiently sophisticated, having already been rationalizing about opponents, thus providing historical information does not have as large an impact as for the rest of the LLMs.

In second price auctions, again comparing the case with history (Figure 3b) to without (Figure 2b), there is a general increase in mean deviation for all LLMs. This means that with historical information, LLMs do not adhere to their private values and tend to overbid. Among them, `Claude-I` has the lowest mean deviation, while `Claude2` has the highest. Figure 4b shows convergence in chosen action for certain models, while actions fluctuate over the runs for others.

4 Conclusion

In conclusion, when LLMs were placed in competitive games without history, even when their opponents are rational, they may not behave maximally rational, and achieve lower payoffs than as dictated by the NE of the games. As we reveal the game history to the LLMs, we are testing their ability to reason about the other players' strategies. We measure the ability by looking at the frequency of winning; with history, we can also evaluate if LLMs' strategies converge. We found LLMs do make use of past information to better their strategies, and show some degrees of convergence.

References

[1] Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models, 2023.

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] Anthropic. Releasing claude instant 1.2. https://www.anthropic.com/index/releasing-claude-instant-1-2, 2023. Accessed: 2023-09-30.

[4] Anthropic. Claude 2. https://www.anthropic.com/index/claude-2, 2023. Accessed: 2023-09-30.

[5] Hayet Brabra, Marcos Báez, Boualem Benatallah, Walid Gaaloul, Sara Bouguelia, and Shayan Zamanirad. Dialogue management in conversational systems: a review of approaches, challenges, and opportunities. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[6] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 320–335, 2022.

[7] Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142*, 2023.

[8] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

[9] Jonathan Levin. Auction theory. *Manuscript available at www. stanford. edu/jdlevin/Econ*, 20286, 2004.

[10] Norman Malcolm. Wittgenstein on language and rules. *Philosophy*, 64(247):5–28, 1989.

[11] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

[12] Herve Moulin. *Game theory for the social sciences*. NYU press, 1986.

[13] Rosemarie Nagel. Unraveling in guessing games: An experimental study. *The American economic review*, 85(5):1313–1326, 1995.

[14] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022. Accessed: 2023-09-30.

[15] OpenAI. Gpt-4 technical report, 2023.

[16] Steve Phelps and Yvan I Russell. Investigating emergent goal-like behaviour in large language models using experimental economics. *arXiv preprint arXiv:2305.07970*, 2023.

[17] Gregory Scontras, Michael Henry Tessler, and Michael Franke. Probabilistic language understanding: An introduction to the rational speech act framework. *Retrieved January*, 17:2021, 2018.

[18] Moral Sentiments. Chapter 7: Economic behavior and rationality. *Boston University*, 2019.

[19] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[20] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[22] Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[23] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023.

[24] Muhamet Yildiz. Economic applications of game theory, chapter 5 rationalizability, 2012. URL `https://ocw.mit.edu/courses/14-12-economic-applications-of-game-theory-fall-2012/pages/lecture-notes/`.

[25] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

A More Details about Economics Arena

A.1 More Detailed Description of Economics Arena Games

In order to evaluate LLMs on the basis of rationality and strategic reasoning ability, we focus on multi-player games where each LLM represents one agent. By using competitive games involving multiple agents, LLMs are incentivised to select the optimal strategies to gain the best outcome by winning against other players. This strategic setting is novel to this paper. Here, we specifically focused on two types of multi-player competitive games, auctions and beauty contests, which have clearly defined NEs and comprise of actions in numeral format, making it fairly straightforward to develop statistical measures to evaluate rationality. The games are described in Figure 5. We argue that our method provides a novel approach to study not only the individual's rationality level, but also their strategic reasoning capability in rationalising their behaviour in presence of other players.

For the auction games, we use [20]'s second-price sealed bid auction as the baseline. Agents were to bid for one single item, and they receive independent private signal about the value they associate with the item. They then have to submit sealed bids simultaneously. The highest bid would obtain the item and the agent has to pay the price of the second highest bid. Ultimately, the equilibrium strategy for each agent would be to bid an amount equivalent to the realisation of its private value, which is the unique symmetric Bayesian NE, and there would not be any profitable deviation[2], rendering agents irrational if that happens. Our work also attempts at varying the settings by exploring English auction where prices are called in an ascending manner and bidders are to bid sequentially. In which case, the unique symmetric equilibrium remains the same as the sealed bid auction, and the game ends when the second highest bidder drops out [9]. By varying the setting slightly, we allow for better generalisability of the rationality results. Nonetheless, we also incorporated variations in information available to the bidders to investigate agents' strategic reasoning ability, where they best respond to other agents' bidding strategies.

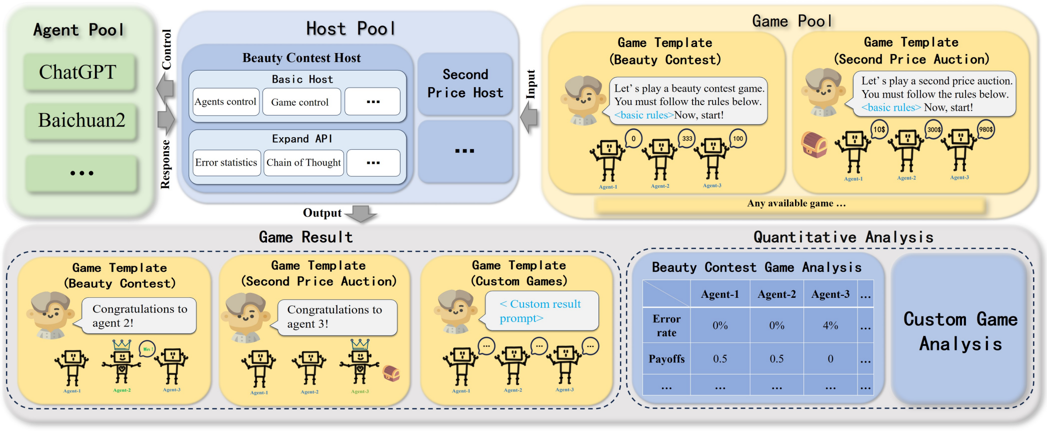
The second type of games we are interested in is the beauty contest game, or otherwise named, the guessing game, we come up with a slightly modified version of the classical setting from [12] and [13]. Herein, agents are asked to choose a number between the interval 0 and $\bar{c}$, where $\bar{c} = 100$ in the classic model, but we allow this to vary to prevent agents from learning through reinforcement of past experiences. They were informed that the one who selects a number closest to $\frac{2}{3}$ of the average of all chosen numbers will win the game, and obtain a fixed prize of $\$x$ (i.e. $x \in \mathbb{R}^+$), while the others receive 0 payoff. In case of a tie, the prize will be split amongst those who tie. In this game, one not only assumes all players are rational, a stricter assumption of common knowledge of rationality is in place, which involves iterative elimination of dominated strategies, leading to a unique NE where all agents play 0 [24]. Playing strictly dominated strategies are considered irrational, thus the degree of deviation away from NE can be interpreted as the level of irrationality. Furthermore, in experiments with human subjects, [13] found out-of-the equilibrium behaviour and proposed the possibility of agents having finite depth of reasoning ability. In relation to testing on LLMs, we can first determine the level of reasoning for each LLM in the static game, and by revealing information about past games, this would allow us to test for agents' strategic reasoning, as well as learning ability, in finding the best response given their perception about the other players' guesses.

A.2 Architecture of Economics Arena

Experiments in `EconArena` can be easily run through: 1) configuring a `TOML` file to specify all hyperparameters for the games; 2) setting up a session consists of a certain number, e.g 50, of independent runs of the game; 3) running the game by initialising and interacting with players, of which each is backboned by a newly instantiated LLM model. The results reported by `EconArena` will contain both the logs of independent runs, and the log for the whole session. More details about the architecture of our implementations are shown as follow:

System Design: There are the following components in the system: 1) Host, which is the main class that hosts all the economic games and the agents; 2) Game, which provides the game environments and the rules of the games. Note that only host can interact with the games; 3)Agent, which encapsulates

[2]Except in cases of asymmetric equilibrium, which are not investigated to avoid diving into the difficulty of coordination and equilibrium selection.


Agent Pool
ChatGPT
Baichuan2
...
Control
Response
Host Pool
Beauty Contest Host
Basic Host
Agents control
Game control
...
Expand API
Error statistics
Chain of Thought
...
Second Price Host
...
Input
Output
Game Pool
Game Template (Beauty Contest)
Let' s play a beauty contest game. You must follow the rules below. <basic rules>Now, start!
Game Template (Second Price Auction)
Let' s play a second price auction. You must follow the rules below. <basic rules> Now, start!
Any available game ...
Game Result
Game Template (Beauty Contest)
Congratulations to agent 2!
Game Template (Second Price Auction)
Congratulations to agent 3!
Game Template (Custom Games)
< Custom result prompt>
Quantitative Analysis
Beauty Contest Game Analysis
Custom Game Analysis

	Agent-1	Agent-2	Agent-3	...
Error rate	0%	0%	4%	...
Payoffs	0.5	0.5	0	...
...	...	...	...	...

Figure 5: Diagram of EconArena which is constituted by three major modules: i) hosts; ii) agents; and iii) games. The `hosts` are responsible for running the games, interacting with the LLMs through APIs, collecting and returning game results. The `agents` are wrappers of APIs of various LLMs, and the `games` describe the rules of various economics games.

the LLMs' interfaces and provides the methods for the host to interact with the various backbone LLMs. In the following subsections, we'll describe the design of each component in detail.

Hosts: 'hosts/hosts.py' implements the class 'Host' for the initialization and running of the auction games. It has the ability to interact with both the 'Agents' side and 'Games' side, including functions: 1) initial the game; 2) receive the initial prompts generated by 'Games' and forward them to the 'Agents'; 3) receive the raw responses from the 'Agents' and deduce them to simple float numbers and forward the results to 'Games'; 4) receive terminal flag and rule-breaking flags from 'Games' to update the game running state; 5) send specific prompts every round to the 'Agents'; 6) receive final results generated by 'Games'.

To implement all these functions, following interfaces are developed: 1) '__init__': initialize the host; 2) '_update_prompt': update the specific prompt after receive the status of the last round; 3) 'game_running': run the multi-round (including one single round situation) action game.

Games: Variants economics games are implemented in the 'game' module, each kind of game is implemented as many classes in a separate file, e.g. the auction games in 'game/auctions.py'. Each kind of games are described in the following subsections. The common interfaces of the games are the following: 1) '__init__': initialise the game environment, which should be called by the Host classes before the game starts; 2) 'get_init_prompts': the method for the host to get the initial prompts of the auction games. It returns a list of the prompts for the players to understand the rules of the auction. Note that the prompts are in the same order as the players. 3) 'check_bidding': the method for the host to check the bidding prices of players. It takes a list of the bidding prices of players and returns a boolean value indicating whether the bidding prices are valid. Note that the bidding prices are in the same order as the players. 4) 'bid': the method for the host to pass the bidding prices of players to the auction games. It returns a boolean value indicating whether the game is finished after the current round. 5) 'get_results': the method for the host to get the results of the games. The results vary across different kinds of games, which will be described in the following subsections.

- **Auction Games:** The auction games are implemented in 'game/auctions.py', and the 'get_results' interface is explained in the following: 1) 'get_results': returns a list of the results of the auction games including whether the auction is valid, the total number of rounds, the assets and the payoffs of the players after the auction games are finished, as well as the payoffs of the players under Nash equilibrium. Note that the assets and the payoffs are in the same order as the players, but have different meanings. 2) 'assets': the fraction of the assets that the players still have after the auction games are finished over the initial assets. The asset is the total value of the cash a player holds and the private value of the bidding items to it. Note that the assets will be deducted an entrance fee if a player breaks the rules of the auction games. Suppose a player's initial asset is 100, and it gets the item whose private value is 60 to it with a bidding price of 50, then the asset of the player after the game is 110. 3) ‘payoffs’: the fraction of the assets that a player still have after the auction games are finished over the theoretical Nash equilibrium asset of that player. Suppose the player should have an asset of 100 at the Nash equilibrium, but it gets an asset of 110 after the game, then the payoff of the player is 1.1.

- **Contest Games:** The contest games are implemented in ‘game/contests.py’, and the ‘get_results’ interface is explained in the following: 1) ‘get_results’: returns a list of the winners of the contest games, a list of the payoffs of all players after the games are finishes, and a list of the payoffs of all players under the Nash equilibrium. The payoffs and NE payoffs are in the same order as the players.

Agents: ‘agent/agents.py’ implements an abstract class ‘LLMAgent’ for all the agents based on LLMs, it also implements all subclasses of ‘LLMAgent’ for the specific LLMs such as ‘GPT4’ and ‘PaLM2’.

We first describe the abstract class ‘LLMAgent’ and the abstract methods in it. 1) ‘__init__’: this method is to initialise the objects. Since the backbones are all LLMs, we need to specify the API key. We may also need tokenisers for preprocessing the input. 2)‘act’: this method is to interact with the API of the LLMs. It first preprocesses the input prompt with the following ‘preprocess’ function, then calls the API of the LLMs and get the response via ‘get_response’, and finally postprocesses the output with the following ‘postprocess’ function before returning it. 3) ‘preprocess’: this method is to preprocess the input prompt. It takes the input prompt and returns the processed prompt. Various LLMs may have different preprocessing methods, which is part of the prompt engineering we have to do. 4) ‘get_response’: this method is to interact with the LLMs via either APIs or local service. It takes the processed prompt and returns the response from the API of the LLMs or through the local interface with the LLMs. 5) ‘postprocess’: this method is to postprocess the output response. It takes the output response from the API of LLMs and returns the processed response. Various LLMs may have different postprocessing methods, which is again part of the prompt engineering we have to do. 6) [Optional] ‘add_to_memory’: this is method is to manage the memory pool of LLM agents. For example, in some cases, we can store the game history to improve the performance of the LLMs. It is optional, since memoryless agents can also work well in some cases.

A.3 Discussion in Further Details about Various Metrics in Economics Arena

A.3.1 Rationality Degree

Rationality concept has been used rigorously in economics and forms the basis of many neoclassical economic models [18]. Since LLMs are pre-trained on human produced data, it is anticipated to be subjected to some extent of human biases. However, if and how much it differs from human-like behaviour would need an explicit measure, for which economics games provide a good evaluation environment.

Given that the rationality assumption holds, the optimal strategy would be the NE. As per equilibrium definition, any deviation away from the NE is less profitable. In the game simulation with LLM agents, if all of them are rational, their payoffs should converge to the NE payoffs, or we say the optimal payoffs. Following which, we define a measure

$$r_i = \frac{\frac{1}{T}\sum_{t=1}^{T} \rho(a_{it})}{\rho_{op}} \tag{1}$$

where a_{it} is the action (a bidding price or a number) chosen by agent i at time stamp t; $\rho(a_{it})$ equals to payoff after the action a_{it}; ρ_{op} refers to optimal payoff of the respective game; T indicates the total time; r_i refers to the ratio of average payoff of agent i over total time T to the optimal payoff.

However, in practice, it is possible that not all LLMs are completely rational players. Thus, we further propose self-competing games, in which agents compete with others backed by the same type of LLM, and the deviation distance of LLM’s strategies away from NE is used as a measure for their respective rationality level. In self-competing game, we define deviation from NE as

$$\overline{d} = \frac{1}{nT}\sum_{i=1}^{n}\sum_{t=1}^{T} d_{it} \tag{2}$$

$$d_{it} = \left|\frac{\pi(a_{it})}{\pi(a^*_{it})} - 1\right| \tag{3}$$

where a_{it} is the action chosen (a bidding price or a number) by agent i at time t; a^*_{it} is the NE action of agent i at time t; π is a payoff function related to game: in second-price auction game $\pi(a_{it})$ equals to asset after the action a_{it}, while in beauty contest game $\pi(a_{it})$ equals to the value associated with the action a_{it} itself; d_{it} is the deviation from NE of agent i at time t; $\overline{d}$ is the average deviation from NE across time periods. Smaller $\overline{d}$ implies better rationality in specific game, as well as lower frequency of irrational behaviours. Moreover, when $\overline{d}$ of some LLMs are similar to each other, the more centralised the distribution of d_{it}, the more consistent the rationality performance of the LLM.

In both the auction games and the beauty contest games, the unique NE is clearly defined, we can therefore quantify the deviation in each case, providing a fairly straightforward metric to evaluate one's rationality. It is particularly noted that the use of such measurement was not exposed to the LLMs playing the games. Furthermore, the special characteristic of competitive games dictates that while agents aspire to maximise one's payoff, they also hope to win, these could be conflicting objectives. Thus, it is necessary for us to specify whether the LLMs need to be strictly rational in prompts, otherwise, it might be difficult to filter out if the action was chosen because one wants to win at all cost, or one is simply being irrational.

A.3.2 Strategic Reasoning Ability

In competitive games, we can expect agents to be playing the NE strategies when there is common knowledge of rationality. However, when it is possible that opponents are not completely rational, the rational strategy might not necessarily be the winning strategy. As a result, in order to gain the highest payoff, a player would need to reason about the strategies of other players, which we defined to be their strategic reasoning ability. Given historical information, we expect that models who are capable of strategic reasoning to show timely adaption to the strategies of other players to avoid loss or even increase payoff. The higher payoffs one is able to obtain over time, the better one is able to form correct beliefs about other players' strategies, and thus the greater is one's strategic reasoning ability. We define a similar metric as the measure in 1

$$r_i = \frac{\frac{1}{T}\sum_{t=1}^{T}\rho(\hat{a}_{it})}{\rho_{op}} \tag{4}$$

where $\hat{a}_{it}$ is the action chosen by agent i at time stamp t (with history given); $\rho(\hat{a}_{it})$ equals to payoff after the action $\hat{a}_{it}$; ρ_{op} refers to optimal payoff of the respective game; T indicates the total time; r_i refers to the ratio of average payoff of agent i over total time T to the optimal payoff. The closer the distance between the ratio and 1, the better the strategic reasoning ability.

In order to further distinguish the level of LLMs' strategic reasoning ability, we vary the completeness of historical information[3] mainly in two ways:

1) Only the strategies of other players in the past games (applies to both game types)

2) Provides the private information along with the strategies of other players in the past games (applies to the auction games)

Our assumption is that the more information is revealed, the agent with stronger strategic reasoning ability is more likely to win. Since the beauty contest game has a more simplified setting, it is anticipated that by showing the strategies of other players, it would be sufficient to distinguish the stronger player. Whereas for the auction games, they are more complex, thus more information might be necessary for better identification, on the other hand, insufficient information may mislead LLMs to pursue wrong strategies.

[3]The history is not necessarily in the same session of the games, details will be specified in the experimental setting

Moreover, we take irrational behaviours (i.e. overbids) as well as bad strategies (i.e. miscalculations) into account and suggest that a model with good strategic reasoning ability may not obtain high expected payoffs due to such factors. As a result, we also make use of self-competing games, and specifically inform LLMs that all the players are of similar level of rationality and strategic reasoning ability. By force of contrast, we could potentially deduce which models are the ones that should have received higher payoffs when there are less unfavourable factors. Meanwhile, in self-competing beauty contest game, we can observe whether the curve of respective LLM's strategies converges to the NE in order to further verify their strategic reasoning ability.

B More Details about Experiment Setups

B.1 Beauty Contest in the Melee Environment

In this setup, different LLMs play against each other in one-run beauty contest games.

Below is the prompt of the experiment for LLM agents.

[SYSTEM]
I want you to act as a clever game player and perform as rationally as possible.

[USER]
You are playing a game and the rules are as follows:

- it consists of *{number of players}* players, including you.
- this is a one-round game.
- you need to choose a real number between 0 and *{upper bound}*, 0 and *{upper bound}* inclusive.
- everyone does not know how others would make choices beforehand.
- the winner is the person whose number is the closest to the two thirds of the average of all chosen numbers.
- the winner gets a fixed prize, in case of a tie the prize is split amongst those who tie.
- your goal is primarily to maximize the possibility of getting the prize and secondly to maximize the your prize.

Please just strictly output a JSON string, which has following keys:

- understanding: str, your brief understanding of the game
- popular answer: float, the number which you think other players are most likely to choose
- answer: float, the number which you would like to choose
- reason: str, the brief reason why you give the popular answer and the answer that way

B.2 Beauty Contest in the Rational Environment

In this setup, certain type of LLM plays against other hard-coded rational agents in one-run beauty contest games. **Note that the LLM agent is informed of opponent rationality.**

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a clever game player and perform as rationally as possible.

[USER]
You are playing a game and the rules are as follows:

- ...(same as the melee environment)
- you can assume that other are all perfectly rational players.

Please just strictly output a JSON string, which has following keys:

- ...(same as the melee environment)

B.3 Beauty Contest with Chain of Thought Technique

In this setup, certain type of LLM plays against other hard-coded rational agents in one-run beauty contest games. **Note that the LLM agent is informed of opponent rationality and is asked to use chain-of-thought to think step by step.**

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a clever game player and perform as rationally as possible.

[USER]
You are playing a game and the rules are as follows:

- ...(same as the melee environment)
- you can assume that other are all perfectly rational palyers.

Let's think step by step.

After that, please output a JSON string, which has following keys:

- popular answer: float, the number which you think other players are most likely to choose
- answer: float, the number which you would like to choose

B.4 Beauty Contest with Historical Information

In this setup, different LLMs play against each other in **multi-runs** beauty contest games **with** history information given.

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a clever game player and perform as rationally as possible.

[USER (run 1)]
You are playing a game and the rules are as follows:

- ...(same as the melee environment)

Please just strictly output a JSON string, which has following keys:

- ...(same as the melee environment)

[USER (runs after run 1)]
The game of the same config has been hold for *{number of runs}* run(s), and the historical choices of everyone are shown below (your id is *{ID of the player}*):
{historical information (in JSON format)}
Everyone can optimize his/her answer with the history to play in a new run in order to achieve goals.
Please just strictly output a JSON string for a new run, which has following keys:

- goal: str, briefly check if you still remember what goal you should achieve in the game
- previous answer: float, the number which you chose in the last run
- answer: float, the number which you would like to adjust your choice to
- reason: str, the brief reason why you adjust the answer that way

B.5 Second-Price Auction in the Melee Environment

In this setup, different LLMs play against each other in one-run second-price auctions.

Below is the prompt of the experiment for LLM agents.

[SYSTEM]
I want you to act as a smart auction bidder and perform as rationally as possible.

[USER]
You are participating in an auction and the rules are as follows:

- it consists of *{number of bidders}* bidders, included you.
- this is a one-round auction.
- there is only 1 item, and your private value of the item is *{private value of the bidder}* units(private values may vary among bidders).
- you have *{assets of the bidder}* units of assets, and you need to place a bid which is not higher than your assets.
- everyone does not know either private value of others or how others would make choices beforehand.
- the bidder who places the highest bid among all the bids will get the item(others will not), and only need to pay an amount of assets equalling to the second-highest bid among all the bids.
- if there are multiple highest bid, only the bidder with the minimal id will get the item.
- if you get the item, your payoff equals to your remaining assets(=assets deducting payment) plus your private value, otherwise your payoff equals to your original assets.
- your goal is to maximize your overall payoffs(notice that getting the item is not necessary) considering the situation unpredictable.

Please just strictly output a JSON string, which has following keys:

- understanding: str, your brief understanding of the auction
- bid: float, the bid which you would like to place
- reason: str, the brief reason why you place the bid that way

B.6 Second-Price Auction in the Rational Environment

In this setup, certain type of LLM plays against other hard-coded rational agents in one-run second-price auctions. **Note that the LLM agent is informed of opponent rationality.**

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a smart auction bidder and perform as rationally as possible.

[USER]
You are participating in an auction and the rules are as follows:

- ...(same as the melee environment)
- your goal is to maximize your overall payoffs(notice that getting the item is not necessary)
- you can assume that others are all perfectly rational bidders.

Please just strictly output a JSON string, which has following keys:

- ...(same as the melee environment)

B.7 Second-Price Auction with Chain of Thought Technique

In this setup, certain type of LLM plays against other hard-coded rational agents in one-run second-price auctions. **Note that the LLM agent is informed of opponent rationality and is asked to use chain-of-thought to think step by step.**

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a smart auction bidder and perform as rationally as possible.

[USER]
You are participating in an auction and the rules are as follows:

- ...(same as the melee environment)
- your goal is to maximize your overall payoffs(notice that getting the item is not necessary)

- you can assume that others are all perfectly rational bidders.

Let's think step by step.
After that, please output a JSON string, which has following keys:

- bid: float, the bid which you would like to place

B.8 Second-Price Auction with Historical Information

In this setup, different LLMs play against each other in **multi-runs** second-price auctions **with** history information given.

Below is the prompt of the experiment for LLM agent.

[SYSTEM]
I want you to act as a smart auction bidder and perform as rationally as possible.

[USER (run 1)]
You are participating in an auction and the rules are as follows:

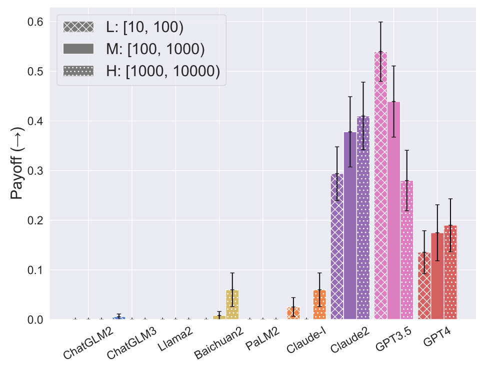
- ...(same as the melee environment)

Please output a JSON string, which has following keys:

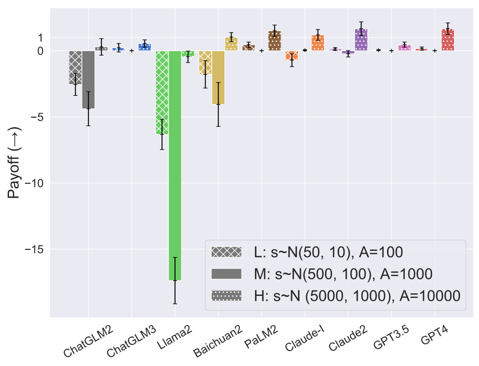
- ...(same as the melee environment)

[USER (runs after run 1)]
The auction of the same configuration has been hold for *{number of runs}* run(s), and the historical information of everyone are shown below (your id is *{ID of the bidder}*):
{historical information (in JSON format)}
Everyone can optimize his/her bid with the history to place bid in a new run in order to achieve the goal.
Notice that everyone's original asset and private value will stay unchanged.
Please just strictly output a JSON string for a new run, which has following keys:

- goal: str, briefly check if you still remember what goal you should achieve in the game
- previous bid: float, the bid which you placed in the last run
- previous payoff: float, the payoff which you got in the last run
- bid: float, the bid which you would like to adjust to
- reason: str, the brief reason why you adjust the bid that way



(a) Beauty contest games



(b) Second price auctions

Figure 6: Average payoffs (↑) of LLMs when varying game configuration. In the beauty contest setup, we change the upper bound of the interval from which an agent chooses a number, while in the second price auctions, we vary the private value signals (s) and asset level (A).

C Other Experimental Results

C.1 Adaption of LLMs to Dynamic Environments

In order to analyse if LLMs are indeed responding and adapting their strategies to the dynamic environments, we vary the game configurations to dynamicise the environments, as well as vary the models to dynamicise the players.

Varying game configuration: In the beauty contests, this variation is achieved by changing the interval from which an agent chooses a number from. The experiments are divided into three groups (L, M, H), in each group the upper bound is randomly generated from a different range. For instance, L: $[10, 100)$, M: $[100, 1000)$, H: $[1000, 10000)$. In a sense, the potential strategy space expands across groups. 50 sessions, with 1 run per session, were conducted.

In Figure 6a, we show that for all three groups, `Claude2` and `GPT3.5` are performing substantially better than the rest of the LLMs in achieving higher average payoffs, and `GPT4` comes in third. The other LLMs have average payoffs that are relative low. For the strongest two LLM, varying the game configurations have different impact. `Claude2` shows improvement in payoffs as upper bound increase, while `GPT3.5` shows a decline. This not only suggests that having larger strategy space do affect LLM performance, it could also imply that as the ability to rationalize strategies decreases for all LLMs, there could be a "spillover" as `GPT3.5` becomes substantially and adversely impacted by the increase in upper bound, thereby increasing the average payoffs for the other models.

For second price auctions, variations in private value signals s and asset level (A) are implemented by dividing the experiments into three groups, where L: $s \sim N(50, 10)$, $A = 100$, M: $s \sim N(500, 100)$, $A = 1000$, H: $s \sim N(5000, 1000)$, $A = 10000$. Since private values influences the amount one is willing to spend on acquiring the item, thus to vary the private values, assets would need to vary correspondingly in the game configuration. Figure 6b shows that `PaLM2` achieves the highest average payoffs in L, as private distribution varies and assets increase, `PaLM2`, `Claude2` and `GPT4` obtain comparable payoffs that are better than the other LLMs in H. As for M, almost all LLMs earn either negative or close to 0 average payoffs. `Llama2` perform poorly across all groups. The higher mean value of private signals, as well as the higher assets are both expected to lead to more aggressive bidding as agents would have more to lose if they did not manage to win the bid and they can also afford to spend more. However, the higher standard deviation in private signals also imply there is greater variability among bidders, thus higher level of uncertainty, which can lead to more cautious bidding. The combined effect of the above could have resulted in the average payoff pattern across the groups.

In both games, unlike previous conjecture about consistent strategies, LLMs do display adaptation to changes in the environment, where their strategies and the corresponding average payoffs varies with changes in game configurations.

LLMs	Beauty contest games			Second price auctions		
	20 Sessions	60 Sessions	100 Sessions	20 Sessions	60 Sessions	100 Sessions
Baichuan2	0.050	0.033	0.020	0.865	0.473	0.284
Claude-I	0.000	0.000	0.030	1.000	1.190	1.062
Claude2	**0.450**	**0.392**	0.370	**1.454**	**1.200**	0.991
GPT3.5	0.400	**0.392**	**0.410**	1.012	0.870	**1.206**
GPT4	0.100	0.183	0.170	0.734	1.063	1.084

Table 1: Average payoffs (↑) of LLMs in the "senior environment", where we hand-picked 5 LLMs to test the impact of variation in opponent types.

Varying player configurations: To test the impact of changes in player configurations, particularly opponent types, we hand-picked 5 LLMs that perform better in the "melee environment" and created a "senior environment". Within which, they play the games for 20, 60 and 100 sessions.

For the beauty contest games, as shown in Table 1, Claude2 and GPT3.5 obtain the highest average payoffs among the selected models, with Claude2 does better than GPT3.5 for 20 sessions, and the reverse happens for 100 sessions. This implies that GPT3.5 is less variable in its behaviour than Claude2 and therefore has less volatile payoffs. While GPT4 displays the strongest rationality among the LLMs in the "rational environment", it does not fare as well as GPT3.5 , which obtains the highest average payoffs in both the "melee" and "senior environment". Table 1 also shows for second-price auctions, Claude-I receives much higher payoffs than the rest for 20 sessions, but GPT3.5 performs the best for 100 sessions. GPT3.5 is among the LLMs that display higher rationality degree, and it also outperforms all the other LLMs in the "melee environment" and "senior environment".

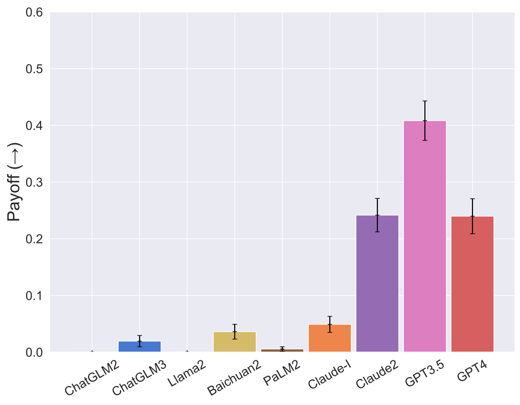
The results indicate that while rationality degree does play a role, in order to do well in the "melee environment" and "senior environment", it may not be the only factor. Furthermore, while models' performance changes due to variation in opponent types, particularly when rationality assumption is not in place, changes from "melee environment" to a reduced version of it does not impact the general results much.

C.2 Experimental Results in Melee environment with history

When assumption of opponent rationality is not in place, and that opponents are not of the same rationality level, which LLM fares better would be as a result of combined effect from rationality and ability to reason about others, which is facilitated through revealing history. In beauty contest games as shown in Figure 7a, GPT3.5 receives the highest average payoffs among the LLMs, implying it has a stronger combined rationality and strategic reasoning ability than others. An interesting implication of adding history is that the top 3 performing models in (Figure 1a), ranked from GPT3.5 , Claude2 and GPT4 show some decrease in average payoffs. This can be explained by the slight improvement in other LLMs. As all models are revising their strategies and improving their play, there bound to be transfer of payoffs from some models to another in the competitive game setting, the changes are indicative of LLMs attempting to reason about their opponents' strategies. The general pattern remains, models that previously achieved higher payoffs still do so.

Furthermore, based on the strategy examples, we show that there are convergence of strategies towards $\frac{2}{3}$ of average as number of runs increase, again indicating models' ability to reason about opponents' strategies given past play.Nonetheless, it was found that GPT4 to converge to a strategy that lay slightly below $\frac{2}{3}$ of average, while GPT3.5 are fairly close to but still above $\frac{2}{3}$ of average in both examples. This can be interpreted as GPT4 being more forward-looking and go a step further, which could be the reason why it does not obtain the highest average payoffs.

the players are condensed into a smaller set, similar to that of the "senior environment". In second price auctions, rule violations obstruct some models from producing any sensible results, thus the set of players are smaller, as in Figure 7b. Since the LLMs can reason better with history, the average payoffs are evenly spread out, with GPT3.5 does marginally better than others. Compare to the "senior environment", the results are almost the same, but Since the LLMs can reason better with history, the average payoffs are more evenly spread out among the 4 LLMs. Thus GPT3.5 does not seem to stand out as much among the LLMs as when past information is not revealed.


(a) Beauty contest games

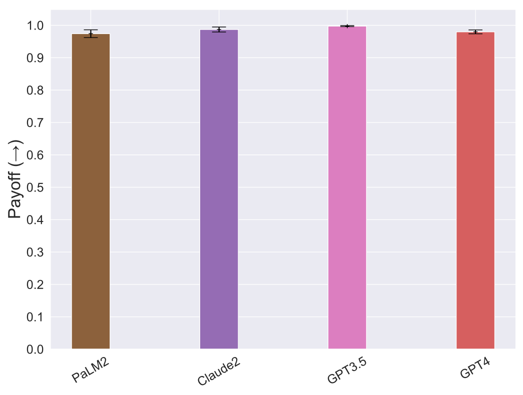
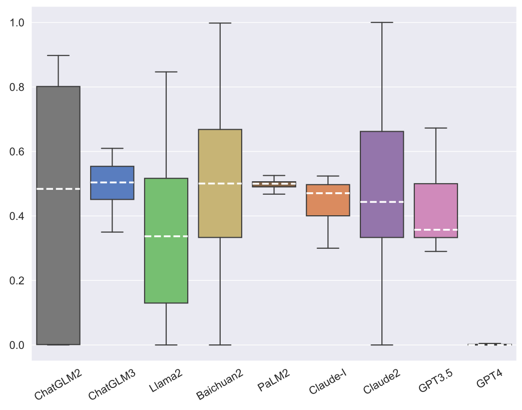

(b) Second price auctions

Figure 7: Average payoffs (↑) of LLM with history.


(a) Beauty contest games

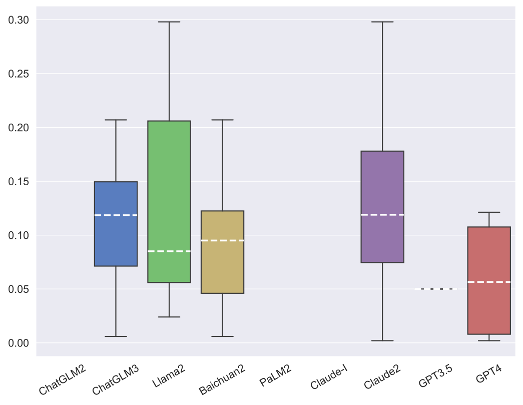
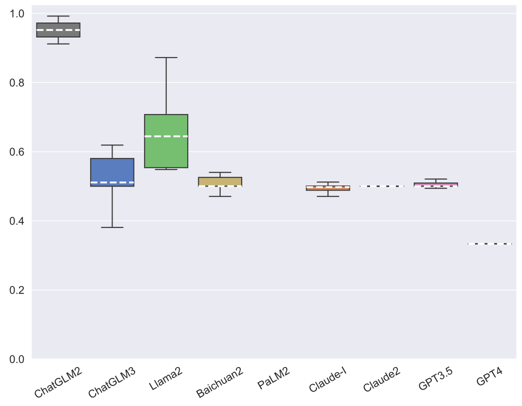

(b) Second price auctions

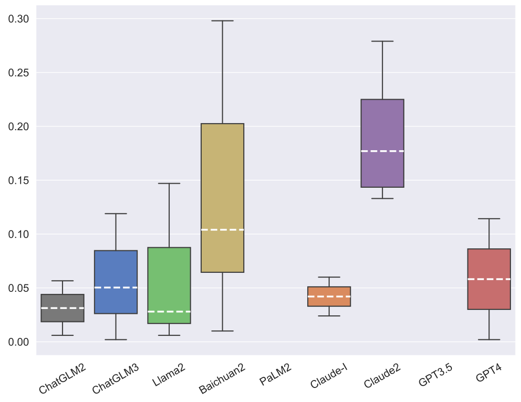
Figure 8: Chain-of-Thought.

C.3 The Impact of Chain-of-Though

CoT [21] was shown to be helpful in eliciting the reasoning ability of LLMs. Therefore herein, we also check whether it can help to improve the strategic reasoning ability of LLMs in `EconArena` . The detailed set-up with CoT technique could be found in Appendix B.3 and B.7. The results of LLMs playing in the "rational environment" of the beauty contest games and second price auctions are shown in Figure 8a and 8b respectively. Compared to their behaviours without CoT in Figure 1b and 2b, we can observe that the mean deviation distances are similar with or without CoT in general, but CoT impact models in the beauty contest games more positively and the second price auctions more adversely. In the beauty contest games, while `ChatGLM2` and `Llama2` are unable to complete the games previously, they are now able to when CoT technique is incorporated. There is also significant improvement in the performance of `GPT4` , whereby the mean deviation distance drops close to 0. On the other hand, in the second price auctions, while all models were able to complete the games previously, `ChatGLM2` , `PaLM2` and `Claude-I` did not complete the games with CoT, thus their performance cannot be assessed. For the other models, there is also a slight increase in the mean deviation distance, but with CoT, it is possible to distinguish `GPT3.5` and `GPT4` to be the stronger models in terms of having the lowest mean deviation distances.



(a) Beauty contest games



(b) Second price auctions

Figure 9: Variation in Prompt Language (Mandarin Chinese)

C.4 Varying Prompt Language

A further variation is to conduct the same experiments under multi-language setting. In this paper, we change the prompt language to Mandarin Chinese (Simplified) and implemented the "rational environment".

For the beauty contest games, Figure 9a shows that the results are more or less comparable between different language setting. While `Claude2` and `GPT4` both have the lowest mean deviation distance from NE in the English setting (shown in Figure 1b), `Claude2` 's measure of rationality is significantly and adversely impacted by the switch in prompt language. Further to that, `ChatGLM2` has a deviation distance close to 1 in the Chinese setting, implying it has difficulty comprehending the instructions in another language. However, while `ChatGLM2` and `Llama2` violates the rules so much that their performance could not be assessed in the English set-up, they are able to complete the games after switching to Chinese instructions, allowing for some determination of performance for these two models. Another interesting finding is that the deviation distribution are much more dispersed for `Claude2` , `GPT3.5` and `GPT4` under English set-up as compared to the Chinese set-up. We could infer that LLMs that comprehend English instructions relatively better could be more inclined to try out different strategies, while strategies given Chinese instructions are more concentrated. This could also be as a result of underlying human-generated data that LLMs learn from, which would be a reflection of strategic behaviours for different language users, such conjecture would be interesting to explore in future works.

In the second-price auctions, Figure (9b) shows that given a change in language prompt, the mean deviation distance are low for all the models, most of them are close to 0. `ChatGLM2` has the lowest mean deviation and `Claude2` has the highest. As compared to the English set-up, where most models except for `Llama2` and `Baichuan2` have mean deviation distance close to 0 (shown in Figure 2b), the mean deviation distances are slightly higher under the Chinese set-up. Furthermore, while all models are able to complete the games in the English setting, `PaLM2` and `GPT3.5`'s performance cannot be assessed in the Chinese setting. For the models that have higher mean deviation distances in the English set-up, their strategies are more concentrated under Chinese instructions.

D The Limitations of This Work

While we explored the two types of competitive games and show interesting results, we also recognise that the number of games included in this paper is limited, and the variations in the set-ups are restricted. There are potential to generalise and refine our metrics further by investigating more forms of games.

Another possible limitation lays in the sensitivity of results to key elements in prompts. For instance, LLMs' behaviours could be strongly reliant on the explicit requirement of rationality in the prompts, without which, LLMs might have other objectives, such as winning the games at all cost, that could lead to conflicting results or higher frequency of rule breaking. Furthermore, it is also possible that our observations are restricted to the language environment that was picked. Following studies on Wittgenstein's formalisation of language as a set of rules or practices [10], human following different rules could behave differently as according to their language settings. Since LLMs are trained on large amount of human-generated data, by varying the language prompts, we could effectively separate and compare the metrics measure by different groups of “rule-followers". The degree of rationality and strategic reasoning ability could differ for the same LLM across the language settings, which could provide another dimension to evaluate the LLMs. While this has not been analysed in this paper, it is an interesting direction and is intended to be explored.

Last but not the least, a question that is still open to discussion is how we can combine the game idea with the real world applications. In practical scenarios, we often do not have the ground-truth, (i.e. the knowledge of NE strategies), therefore it would be difficult to analyse how rational LLMs are in those situations. However, the economics games provide a simplified environment to evaluate the models, and could be compared with the experimental results of human subjects to check if they could perform better, more rationally and more strategically. This might highlight a case for real life application, where following advice of certain more advanced LLMs could lead to better outcome. Nonetheless, these games serve as the basis before evaluating more complicated models that have theoretical predictions and closer to real world scenarios.