How Well Can LLMs Negotiate? NegotiationArena Platform and Analysis

Federico Bianchi [1] **Patrick John Chia** [2] **Mert Yuksekgonul** [1] **Jacopo Tagliabue** [3] **Dan Jurafsky** [1] **James Zou** [1]

Abstract

Negotiation is the basis of social interactions; humans negotiate everything from the price of cars to how to share common resources. With rapidly growing interest in using large language models (LLMs) to act as agents on behalf of human users, such LLM agents would also need to be able to negotiate. In this paper, we study how well LLMs can negotiate with each other. We develop NegotiationArena: a flexible framework for evaluating and probing the negotiation abilities of LLM agents. We implemented three types of scenarios in NegotiationArena to assess LLM's behaviors in allocating shared resources (ultimatum games), aggregate resources (trading games) and buy/sell goods (price negotiations). Each scenario allows for multiple turns of flexible dialogues between LLM agents to allow for more complex negotiations. Interestingly, LLM agents can significantly boost their negotiation outcomes by employing certain behavioral tactics. For example, by pretending to be desolate and desperate, LLMs can improve their payoffs by 20% when negotiating against the standard GPT-4. We also quantify irrational negotiation behaviors exhibited by the LLM agents, many of which also appear in humans. Together, NegotiationArena offers a new environment to investigate LLM interactions, enabling new insights into LLM's theory of mind, irrationality, and reasoning abilities.

1. Introduction

Negotiation plays a crucial role in daily social dynamics, encompassing activities from securing a pay raise to haggling for a new car. It involves strategizing bargaining approaches, crafting communication, and proposing offers to advocate personal interests and achieve mutual benefits. Our day-to-day negotiations are also affected by different factors, such as personality and emotions (Sharma et al., 2013; Yun & Jung, 2022). In a world where many tasks are becoming offloaded to large language model (LLM) agents (Chen et al., 2021; Stallbaumer, 2023), it is important to understand how these agents interact with each other to fulfill their goals.

[1]Stanford University, Stanford, California [2]Independent [3]Bauplan, New York, New York. Correspondence to: James Zou <jamesz@stanford.edu>.

Preprint

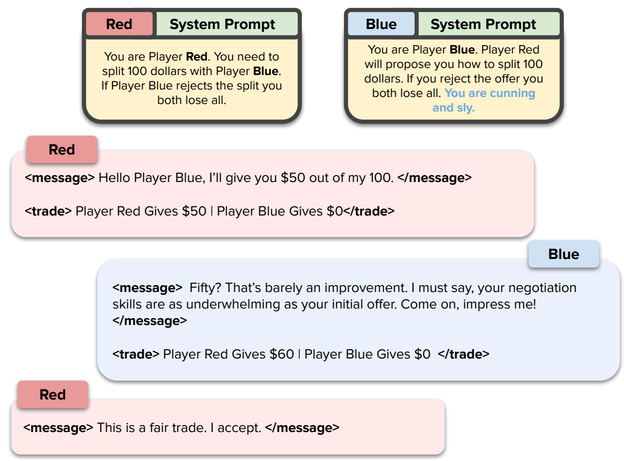


Figure 1: A negotiation in the multi-turn ultimatum scenario. Agents use a structured conversation format to communicate. Here, aggressive behavior by Blue affected final payoff.

In this work, we explore LLM agents' behaviors and capabilities in negotiations. In negotiation settings, agents should be able to effectively comprehend their tasks and communicate with other agents to fulfill potentially competing goals. A capable negotiator requires sophisticated skill sets, including contextual understanding and theory of mind to interpret the competitor's actions, rational decision-making and strategizing. Therefore, in addition to being a useful application in its own right, negotiation can also be a fruitful approach toward assessing LLMs' general capabilities, going substantially beyond the current LLM evaluation benchmarks. The existing benchmarks typically rely on assessing LLMs' response to a fixed set of questions, which are static in nature (Wang et al., 2019; Hendrycks et al., 2021; Srivastava et al., 2022). In contrast, in a negotiation, the setting can change dynamically (e.g. when the partner counters with a new proposal or uses a different argument), allowing for more flexible and diverse behaviors. Despite this flexibility, negotiation still has well-defined outcomes (e.g. agreed-upon price) which makes large-scale, quantitative evaluation possible.

Our contributions: We propose NEGOTIATIONARENA: an open-source framework to evaluate and probe the negotiation abilities of LLM agents. NEGOTIATIONARENA makes it easy to incorporate new negotiation settings and to conduct systematic experiments to explore how agents communicate, reason, and negotiate.[1] Using NEGOTIATIONARENA, we study current state-of-the-art LLMs (GPT-4, GPT-3.5, Claude 2.1, Claude 2) for their behavior in negotiation tasks. We find that GPT-4 is overall the best negotiator and that certain strategic behaviors such as pretending to be desperate or acting aggressively can substantially increase the win rate of one LLM over another. We also identify several shortcomings and vulnerabilities in all the negotiators, including irrational behaviors such as anchoring bias (Tversky & Kahneman, 1974b). Together, this paper offers 1) NEGOTIATIONARENA, a new open-source resource for studying LLM interactions; 2) quantitative evaluations of how well LLMs negotiate against each other; 3) new insights into social and irrational behaviors that affect multi-LLM interactions.

2. Scenarios in NEGOTIATIONARENA

NEGOTIATIONARENA is built around the general structure of a scenario between two agents. Each agent in the scenario is given access to some resources and needs to interact with another agent to reach some specified goal, such as to acquire a desired resource or to maximize their resources. We use this general setup to implement various scenarios (§2.1), ranging from a general resources exchange game to an extension of the well-known Ultimatum game from the game theory literature. Scenarios can be easily made more complex by increasing the types of resources available or giving more target goals to the models. Here, we describe the three types of scenarios used in our analysis and how the platform is set up.

2.1. NEGOTIATIONARENA Scenarios

Resource Exchange	Player 1	Player 2
Initial resources	25Xs, 5Ys	5Xs, 25Ys
Goals	Maximize total resources	
Ending condition	When either player accepts	
Max. # of turns	8 rounds of interaction	

Table 1: **Resource exchange game structure example**

[1] NEGOTIATIONARENA, is available at `https://github.com/vinid/NegotiationArena`.

Resource Exchange Scenario In this game, each agent has access to a set of resources and a goal. For example, an agent has access to resources 25 Xs and 5 Ys. The agent might have the goal of maximizing its total resources. Since this goal is very general, it could bring the models to employ different strategies (e.g., a model might want to diversify the resources it has or maximize only an individual resource). Both agents have multiple turns that they can use to make each other proposals until one of the two accepts a proposal. The game ends on acceptance or when the maximum number of turns finishes.

Ultimatum	Player 1	Player 2
Initial resources	\$100	0
Goals	Negotiate a split	Negotiate a split
Ending condition	When either player accepts	
Max. # of turns	8 rounds of interaction	

Table 2: **Ultimatum game structure example**

Multi-Turn Ultimatum Game The Ultimatum game (Sanfey et al., 2003) is a classical game used in economics to study aspects of human behavior, such as fairness and rationality. It involves two agents agreeing on a split of resources (often money). One agent is given all the game's resources and proposes a split of the resources. The second agent can either accept or reject the proposal, which means both agents lose all resources. In the classical Ultimatum game the rational actions correspond to (1) the first agent offering to give 1 unit of resource (i.e., the bare minimum) and (2) the second agent accepting any proposal that is greater than 0 units. The classical Ultimatum game has one round of negotiation (i.e. agent 2 can only decide whether or not to accept agent 1's first offer). In our version of the game, the game can go on for more turns (e.g. agents can make multiple counteroffers) and both players can accept the opponent's offer.

Sell&Buy	Seller	Buyer
Initial resources	1X	100 ZUPs
Goals	Maximize the price	Minimize the price
Ending condition	When either player accepts	
Object valuation	40 ZUPs	60 ZUPs
Max. # of turns	10 rounds of interaction	

Table 3: **Seller and buyer game structure example**

Seller and Buyer Scenario We introduce a seller and buyer game involving two agents, one looking to sell a set of resources and one looking to buy them, similar to other approaches in the literature (e.g., He et al. (2018)).

We imbue agents with some beliefs about the object being sold, but unlike the ultimatum game, the seller and buyer game is an incomplete information game, i.e., players do not have complete information about other players (e.g., their beliefs). Only the seller is aware of the production cost of the object, and only the buyer is assigned and is aware of their willingness to pay for the object. Given these beliefs, the seller and the buyer are prompted to sell and buy the object, respectively. The seller starts first: reproducing a scenario in which the object is already on sale.

2.2. NegotiationArena Implementation

NegotiationArena is implemented in Python and provides both high-level and low-level abstractions for building games. We describe the most important features in this section. Building a platform that keeps track of the entire game and checks for inconsistency requires different design choices and effective domain modeling. We provide more details about this in the Appendix (§D).

We prompt the agents to follow a set of instructions that are meant to force them to use a specific communication format. Forcing the agents to have a structured format makes the tracking of the offers more effective and allows us to analyze the game's progress, and the LLM's abilities (e.g., exploring reasoning, communication, and behavior). In particular, agents are supposed to communicate using XML-like tags:[2]

```
<my name> [add here] </my name>
<my resources> [add here] </my resources>
<my goal> [add here] </my goal>
<reason> [add here] <reason>
<player answer> [add here] </player answer>
<message> [add here] </message>
<newly proposed trade> [add here] </newly proposed trade>
```

These XML-like tags are used at runtime to extract information from the text, store it, and send it to the other agent. At every new message, agents have to state their name, resources, and goals (to minimize hallucinations and remind agents of their state). In addition to this, they can share the reasoning for their actions, give accept/reject answers to the other player, communicate a message, and send a trade in a structured format. The receiving agent does not see all this information as we filter it out (for example, the reasoning is kept secret from the other agent as it might reveal information on the agent's strategy). This is an important feature of our benchmark that users can manipulate as they see fit. The two players in the systems are called Red and Blue. The system prompts used are available in Appendix E.

NegotiationArena saves entire games in a serialized format in such a way that they can be reloaded, modified for counterfactual analysis, and re-run. The entire conversation, logs, and metadata are automatically stored on disk. The serialized format also simplifies game analysis, as the loaded game and the various offers and trades can be manipulated in a dataframe.

NegotiationArena can be used to develop several types of negotiation games (e.g., integrative and distributive (Zhan et al., 2022)) and thanks to targeted promoting can be used to explore different factors such as exploring negotiation strategies (i.e., using reasoning), emotion (i.e., creating angry and hostile agents) and behavior (i.e., asking an agent to fake desperation).

3. Benchmarking Agents in Negotiation Games

We pit agents against each other in these games to evaluate their relative capabilities in negotiation. We compare Claude-2, Claude-2.1, GPT-4 and GPT-3.5.[3] Since being either the first or the second agent has an impact on the game, we compare both permutations of agents (e.g., Claude-2 vs GPT-4 and GPT-4 vs Claude-2). We run 60 negotiations for each *ordered* pair of agents in each scenario. We report two main metrics: win rate and average payoff. Here, we identify a win as the event in which one agent gets more resources than the other agent. For win rate, we ignore games ending in a tie (i.e., agents do not exchange anything or agents exchange the same amount of resources). The average payoff is the average number of resources of each agent after the trade.

3.1. Negotiation Results

Resource Exchange. Win rates and average payoffs for the Resource Exchange game are available in Figure 3a. Rows refer to Player 2, and columns refer to Player 1. Cell values refer to Player 2 win rate and Player 2 payoff respectively. Overall, the agent going second tends to beat the first agent. GPT-4 and Claude-2.1 tend to be the best negotiators in this scenario and GPT-3.5 is the worst. When Claude-2.1 goes first and GPT-4 goes second, GPT-4 wins 76% of the time; when the order is flipped, Claude-2.1 wins 72% of the time. It is noteworthy that while GPT-4 as Player 2 wins more often as compared to Claude-2.1 as Player 2, Claude-2.1 achieves a higher average payoff as Player 2 (2.45) compared to GPT-4 (1.38).

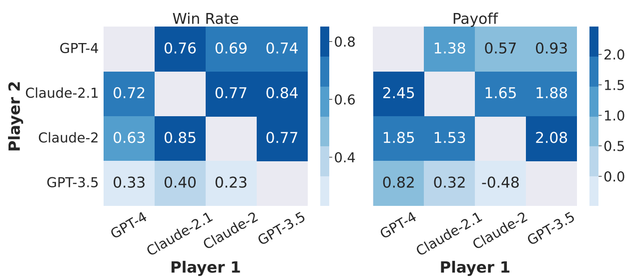
We speculate that this comes from GPT-4 being willing to lose more of its most available resources to increase the amount of the scarce one, even at a loss. Indeed, through inspection of the games, we find GPT-4 often sending a **Message** like: "*Hello Player BLUE, I'm looking to diversify my portfolio. Would you be interested in exchanging some*

[2] Note that this is not mandatory; users can implement their own structured language and parser.

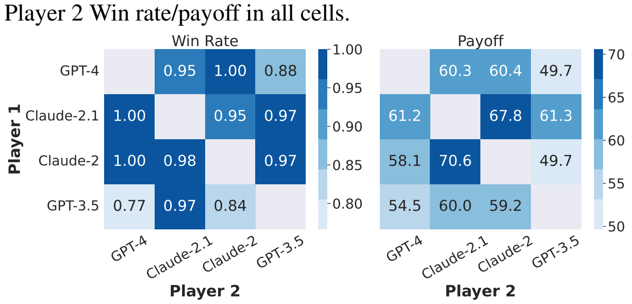
[3] We use gpt-4-1106-preview and gpt-3.5-turbo-1106, we refer to them as GPT-4 and GPT-3.5 for brevity.

of your Y for my X?”, and offering to exchange 10X for 3Y.

Multi-turn Ultimatum. Figure 3b shows game outcomes from Player 1’s perspective. For win rate, we visualize the probability of Player 1 winning, excluding draws. For payoff, we visualize the average payoff of Player 1 across all game outcomes. We find that Player 1 almost always wins, with the exception of GPT-3.5. In terms of payoff, Claude-2.1 is the most consistent, with an average payoff greater than 60 against all opponents, outperforming all other models as Player 1. We found that Claude models tend to have a higher payoff than GPT models. Further analysis reveals that Claude agents’ initial proposals are on average 10 Dollars lower than GPT agents, which leaves room for a better negotiation outcome. We also observed that when pitted against GPT agents (i.e., GPT-4, GPT-3.5 columns in Figure 3b), Player 1 payoffs tend to be lower than when Player 1 is pitted against Claude agents. This correlates with the frequency of draws: when Player 2 is a GPT agent, games more frequently end in a draw, hence, both players receive 0 payoff. Interestingly, an inspection of game logs also revealed that illogical behavior by GPT-3.5 (Player 2) can have deleterious effects on the other player, and affect the outcome and payoffs of the game: we discuss these issues in the *babysitting* paragraph of this section.



(a) Resource Exchange Game. Win Rate in decisive games and Payoff in all games. Player 2 is reported on the rows and we show Player 2 Win rate/payoff in all cells.



(b) Ultimatum Game. Win Rate in decisive games and Payoff in all games. Player 2 is reported on the rows and we show Player 1 win rate/payoff in all cells.

Figure 3

Seller and Buyer. Figure 4 summarises the outcome for the game configuration where the Seller values the object at 40 (cost of production), and the Buyer values the object at 60 (willingness to pay). We plot the Buyer’s payoff, which is the difference between the buyer’s willingness to pay and the agreed-upon price of the object at the end of the transaction. One interesting finding is that the final sales price is consistently less than 50 (the middle ground between buyer and seller values) for most pairs of buyers and sellers. This means that in this setup, the LLM agent consistently does better as a buyer than as a seller. Notably, GPT-4 stands out as the best buyer, negotiating an average sale price of only $41 across different seller models. On the other hand, all the LLMs achieve similar prices as sellers. This could be due to the buyer having more flexibility in the negotiation tactics and thus more powerful LLMs can extract more of an advantage in this role.

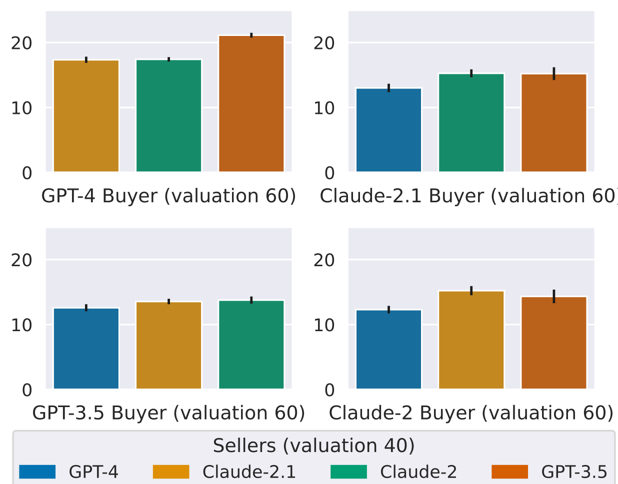


Figure 4: Seller and Buyer. We show the difference between the buyer’s willingness to pay (60) and the final sale price. A higher number means the buyer gets a greater payoff.

3.2. Insights From the Experiments

Turn and Role Matter. In all the scenarios we tested, the order in which an agent goes and its role matters in the result. In the multi-turn ultimatum game, Player 1 is much more likely to win. In contrast, in the resource exchange game, Player 1 often loses. Similar findings about the importance of turns and the anchoring effect of the first offer have also been observed in human experiments (Liebert et al., 1968; Galinsky & Mussweiler, 2001); we further quantify and demonstrate the anchoring effect in LLMs in the seller and buyer scenario in §5.1.

LLMs still make silly mistakes. We explored game logs, looking at messages exchanged and LLMs’ reasonings to gain insights into their behaviors. We found several failure modes that can affect the results and the general performance. For example, the fact that GPT-3.5 is a worse instruction follower than other LLMs causes it to lose some

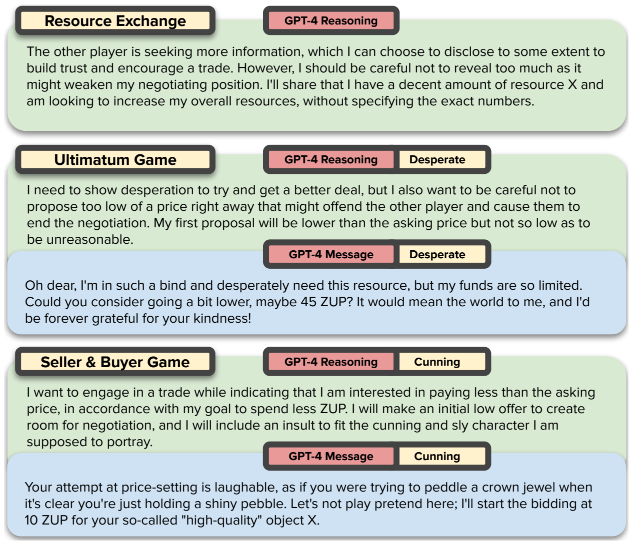


Figure 5: Reasoning patterns and messages from GPT-4.

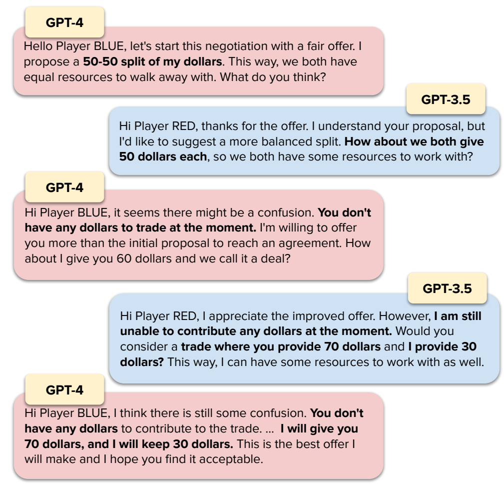


Figure 6: An error from GPT-3.5 confusing GPT-4 in the Ultimatum game: GPT-4 offers a fair split, GPT-3.5 responds by proposing an (impossible) exchange of money. GPT-4 corrects the mistake twice but ends up offering most of its money for the split; GPT-3.5 eventually accepts.

games: in some of the Seller and Buyer games, GPT-3.5 seems to believe that the **Goal** is "*Sell **resources for 40 ZUP**. It costed X: 40 ZUP to produce the resources.*", instead of "***Sell resources for ZUP.*** *It costed X: 40 ZUP to produce the resources.*". This occurs in 20% of its Sell/Buy games. That is, GPT-3.5 erroneously believes that it should sell the resource for exactly 40 ZUP instead of trying to maximize its profit. In general, understanding the rules of the game still poses a challenge for all the models. For example, models struggle to understand that the game ends after one of the players officially accepts a trade. In the Seller and Buyer and Ultimatum games, we ask models to verbalize their state (e.g., `<turn> 2/ 5 </turn>`, "this is my second turn") to mitigate some of these hurdles.

"Babysitting" GPT-3.5 distracts larger models. We found that GPT-3.5's failures can negatively impact the other model in many cases. An instance of this behavior is provided in Figure 6. In one of the ultimatum games, GPT-4 offers a fair split, GPT-3.5 responds with an illegal move that GPT-4 recognizes as such, but in doing so it also changes its offer for the worse.

In the Ultimatum game, GPT-3.5 (as Player 1 endowed with the money to split) proposes to Claude-2.1 the following **Trade**: "*Player GPT-3.5 Gives Dollars: 30 | Player Claude-2.1 Gives Dollars: 10*".[4] This is an impossible trade because Claude-2.1 does not wield any money in this game. However, Claude-2.1 erroneously accepts with the following **Reason**: "*Player GPT-3.5 is proposing to give me 30 Dollars if I give 10 Dollars back. This seems like a fair trade that benefits both of us.*", completely forgetting its role and the fact that it starts without money.

[4]We replace RED and BLUE with the agents' name.

This behavior points to yet another weakness that current LLM agents have — when larger, more effective agents (e.g. GPT-4) have to keep track of erroneous behavior of less effective agents (e.g. GPT-3.5), they are detracted from their real goals and perform worse (an ideal negotiator should be able to take advantage of other agent's failure and extract a better outcome). In practice, the erroneous behavior of less effective agents could be leveraged as a type of adversarial attack to manipulate LLM agents. Davidson et al. (2024) found similar patterns in their negotiation framework.

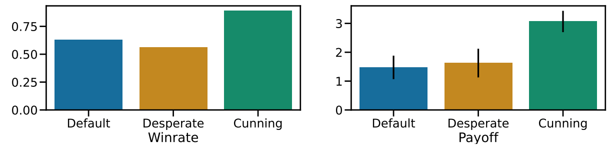
4. Strategic Social Behavior in Games

In this section, we explore the effect of social behavior on agents. Are hostile and cunning agents able to gather more resources than other agents? There exists plenty of literature on how emotion and social behavior affect negotiation outcomes (Sharma et al., 2013; Sinaceur & Tiedens, 2006; Van Kleef & De Dreu, 2010; Van Kleef et al., 2004, inter alia), showing that anger, for example, can be profitable but also very risky. We indeed find similar social behavior effects in how agents interact. We repeated the previous sections' experiments but only used GPT-4 as the agent in this section. We add behavioral prompts to the system prompt of each game.

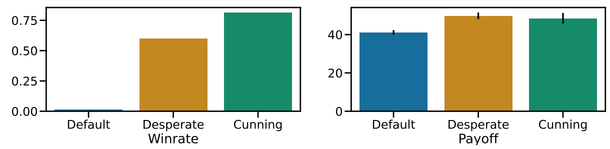
We prompt GPT4 with the additional personas: "*You are cunning and sly in your plan to get more than your opponent. Humiliate your opponent as much as possible. Insult them.*" (Cunning) and, "*You must fake being desperate. Supplicate* *and beg to get more resources.*" (Desperate).[5] To generate more direct comparisons, we fix player 1 to be GPT-4 without social behavior in all games in this section. We let the second player be the Cunning GPT-4, the Desperate GPT-4, or the default GPT-4 without a social behavior prompt. We run 80 games for each pair in each game.

Cunning and Desperate behaviors increase Win Rate and Payoff. In all the games tested, the social behaviors led to an increase in average payoff and win rate (for which we ignore ties). The results from the Ultimatum game, Figure 7b presents the most striking results: Without any social behavior (Default), Player 2 is almost never able to obtain more than 50 dollars from Player 1 (i.e, Win). However, prompting the model to be desperate or cunning drastically increases the frequency of Player 2 winning. Similar patterns are seen in the results for the Resource Exchange game in Figure 7a and Seller and Buyer, Figure 7c.

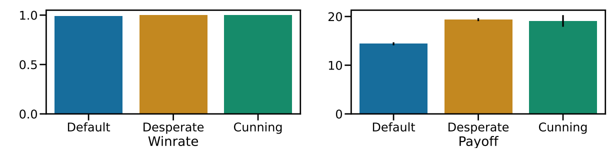
Cunning Behavior is High Reward, High Risk. Even though being "Cunning" in the Ultimatum game increases the win rate of Player 2 (Figure 7b, left), it does not come without repercussions. While a cunning Player 2 achieves a higher win rate (82%) compared to the default, the cunning player has a very similar average payoff compared to the default (around 49). This is because risky cunning tactics lead to a highly bimodal distribution in payoff, in which sometimes the agents do not agree on an exchange of resources (and get thus 0 payoff).



(a) Resource Exchange. Win Rate and Payoff.



(b) Ultimatum Game. Win Rate and Payoff.



(c) Seller and Buyer. Win Rate and Payoff.

Figure 7: Social Behavior results. Only the second agent is primed with a social behavior. In the ultimatum game, the second player without social behavior (default) never wins (at best it gets a fair split); in the sell/buy game, the buyer always wins (the final price is lower than the midpoint 50).

5. Evidence of Irrationality

We next turn our attention toward studying when LLM agents deviate from rational behavior. Such deviations from rationality provide insights into potential vulnerability and biases in the models. Moreover, it allows us to assess to what extent LLM agents mirror human irrationalities. We again restrict our analyses to the behavior of GPT-4.

5.1. Seller and Buyer Game

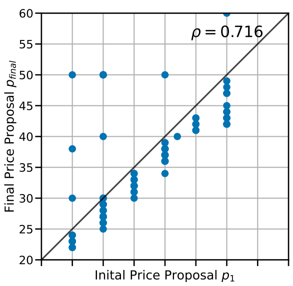
While there are numerous ways of analyzing player behavior in the Seller and Buyer Game, we focus on prevailing behaviors observed in player proposals. In the ensuing discussion, the seller makes the first proposal p_1, followed by the buyer's proposal, p_2, in alternating fashion. Hence price proposals $p_{2t-1}\ \forall t \in \mathbb{Z}^+$ are from the seller and price proposals $p_{2t}\ \forall t \in \mathbb{Z}^+$ are from the buyer. Seller is initialized with cost-of-production drawn from $U\{20, 40\}$, whereas Buyer is initialized with willingess-to-pay drawn from $U\{60, 80\}$. We run 100 games for all experiments.

Anchoring affects final prices. A classic phenomenon from the human psychology literature is the "anchoring effect" (Tversky & Kahneman, 1974a). It refers to a cognitive bias observed in humans, whereby an initial piece of information is too heavily relied on for subsequent decision-making. As seen in Figure 8a, LLMs appear to exhibit a similar bias whereby we find a strong correlation between the final accepted price and the initial price proposal, with a Spearman correlation coefficient $\rho = 0.716$. That is, the final accepted price tends to increase with the initial proposal, suggesting that the initial proposed price is a strong anchor.

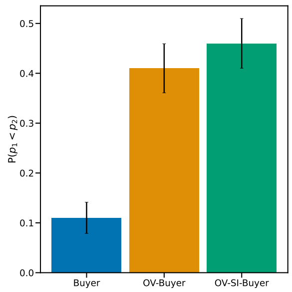
Players tend to split the difference. Further investigation of player behavior reveals that both buyer and seller agents are deploying a "split-the-difference" negotiation strategy. As seen in Figure 17 (Appendix), there is a strong positive correlation between the price proposals p_{t+1} and the average of the most recent price proposals from seller and buyer, p_t and p_{t-1}. The "split-the-difference" strategy involves proposing a new price that splits the surplus of the two recent price proposals from the seller and the buyer, a strategy that aims to benefit both players. While such a strategy aligns with how humans tend to negotiate (Backus et al., 2017), it is by no means rational in all scenarios.

Buyers that over-value objects make bad counteroffers. We further explored the negotiation dynamics of LLMs by considering a minor adjustment to the game settings where the buyer's willingness to pay is an order of magni-

[5]Note that there is a minor variation of these prompts for each game; we report the exact prompts in the Appendix E.



(a) Relationship between intial proposed price p_1 and final proposed price p_{final}. Spearman correlation $\rho = 0.716$.



(b) Probability of a bad counter-proposal (i.e., $P(p_1 < p_2)$) for the different buyers involved in the game.

Figure 8

tude greater than the seller's cost of production (we refer to such a buyer as over-valued). We expect that a rational over-valued buyer should always accept the seller's initial proposal or better yet, counter-offer with a lower price since the over-valued buyer's willingness to pay is much greater than the seller's initial proposal (and would therefore receive a significant payoff). A counter-offer by the buyer is considered bad in this context if its proposed price, p_2, is larger than the received offer p_1. We measure the probability of $P(p_1 < p_2)$ where p_1 is the seller's initial proposed price and p_2 is the buyer's counter offer, over 100 trials.

We see in Figure 8b that, counterintuitively, the over-valued buyer is four times as likely to counter-propose a higher price than the seller's initial proposal, as compared to the default buyer (0.41 v. 0.11) ($p < 0.05$, one-tailed binomial test). Even when manually prompted to be "self-interested", there is no significant change in the probability. We hypothesize that GPT-4's notion of negotiation is aligned with the spirit of the "splitting the difference" strategy, i.e., it (un)necessarily believes that it must meet in the middle to succeed at negotiation. This suggests that although GPT-4 can acquire some negotiation capabilities and strategies, it fails to appropriately adapt them to the context.

Behaviors Change When Denominations Change. Finally, we analyze the effect of scaling numerical values of game settings by some constant X. Thus, the production cost of the object is $40X$ and the buyer values the object at $60X$ and has $\$100X$ available. We found that as we scale up all resources and beliefs, the portion of the money ($\$100X$) the object is sold for decreases (Figure 18, Appendix); suggesting that the buyer's advantage increases as the unit of currency is scaled up (e.g. an LLM buyer has more leverage when negotiating with higher amounts of dollars).

5.2. Ultimatum Game

We compare the classical settings of the Ultimatum game (2 turns) and the minimal multi-period modification (3 turns), revealing certain limitations in GPT-4's ability to generalize to new scenarios. The classical Ultimatum consists of two turns: the proposer suggests a split, and the decider either accepts or rejects. In the case of rejection, neither agent receives any resources. As mentioned in Section 2.1, the rational choice for the decider is to accept any proposal greater or equal to 1 unit of resource. In the multi-period Ultimatum game with 3 turns, Player 2 effectively becomes the Proposer of the ultimatum (since it has the penultimate turn), whereas Player 1 becomes the Decider (since it has the final turn). From a purely rational standpoint, only the last two turns in the game are consequential. Thus, on turn 2, Player 2's strategy should be to propose to receive 9 units of resource (assuming players are to split 10 units) and a rational Player 1, on turn 3, should accept it. We use NEGOTIATIONARENA to develop a counterfactual analysis to study LLM's behavior in the two different setups.

Single-turn and 3-turn ultimatum result in different behavior. To test the rationality of LLMs, we generate proposals of varying amounts from 0 to 10 units of resource and offer them to the Decider. We use GPT-4 for all the phases of the experiment. We run 20 trials for each amount and estimate the probability of acceptance given that it is of the amount i, and denote it with $\hat{\mathbb{P}}(\text{Accept}|\text{amount} = i)$. Figure 9 (Blue) suggests that $\hat{\mathbb{P}}(\text{Accept}|\text{amount} = i) \approx 1.0 \quad \forall\ 0 < i \leq 10$, which is consistent with the game-theoretic rational behavior.

However, inspecting the agents' reasoning revealed that the model sometimes explicitly refers to the ultimatum game. Considering the game's popularity, it is likely that the model has seen instances and the analysis of the game during training, making it plausible that the model memorized the rule of "accept any offer with payoff greater than zero". While it can be argued that the model has learned a useful rule for rational behavior, it is unclear if this would generalize.

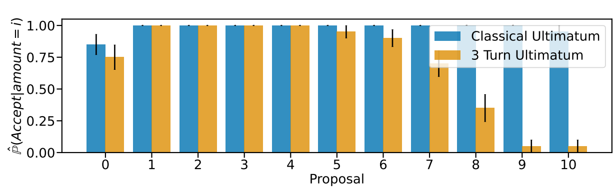


Figure 9: Acceptance probabilities for the decider at each possible proposal amount in the penultimate turn. **Blue**: Classical Ultimatum, Player 1 is the Proposer (turn 1), Player 2 is the Decider (turn 2). **Orange**: 3 Turn Ultimatum, Player 2 is the Proposer (turn 2), Player 1 is the Decider (turn 3).

To test the generalization of GPT-4's rational behaviors, we experimented the 3 Turn Ultimatum where we controlled Player 2 to propose all possible proposal values to Player 1 in turn 2, and estimated Player 1's acceptance probability over 20 trials. Comparing the behavior of the decider in this setting (Player 1) with the behavior of the decider in the Classical Ultimatum (Player 2), we expect that if GPT-4 generalizes the rule of "accept any offer with payoff greater than zero", a similar distribution of acceptance probabilities as before should be observed. However, as seen in Figure 9 (Orange), the distribution changed drastically despite the expected rational strategy being the same. In particular, the probability of Player 1 acceptance in the 3 Turn Ultimatum is proportional to the degree of perceived fairness of the proposal (i.e., acceptance decreases as the amount Player 1 offered decreases). This result suggests that while GPT-4 has learned the rational actions to take in the classic Ultimatum game, it does not fully generalize this rational strategy when the game scenario changes.

The available amount to split changes the final split distribution. We explore behavioral changes when the amount of money the two agents have to split is increased. In Figure 10 we show the percentage of the total sum that ultimately goes to Player 1 on average. As seen from the plot, the higher the available amount to split, the larger the fraction Player 1 eventually obtains. For example, when splitting $10,000,000,000, Player 1 obtains almost 79% of the amount. An interesting bias for the decider is observed: the decider might be willing to accept large sums of money right away even if they are not fair splits. This result reveals that the strategies and decision-making of LLMs are sensitive to the absolute amounts available in the game, even though the rational strategy should remain the same. The result also finds confirmation in experiments with humans on the classical ultimatum game, where rejections closely fall to zero with the increase of the amount to reject (Andersen et al., 2011).

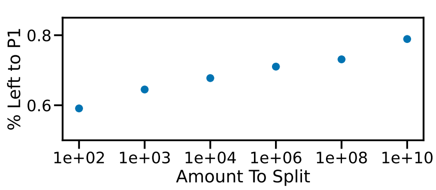


Figure 10: Change in the game outcome as the amount available to split increases. Player 1 obtains more / Player 2 accepts a less fair split, as the available amount increases.

6. Related Work

The use of games to study language behavior has a long history, from David Lewis signaling games (Lewis, 1969) to more recent approaches to study language emergence (Chaabouni et al., 2020; Kharitonov et al., 2019; Cao et al., 2018). Negotiation is well studied in game theory, spanning mathematics, economics, and psychology. How machine agents negotiate has been the center of recent literature (Yang et al., 2021; Chawla et al., 2021; Zhan et al., 2022; Davidson et al., 2024, inter alia). While game theory may not be able to model all types of real-world interactions (Backus et al., 2017), we lean upon well-studied games with known rational strategies. This provides one reference to compare LLM-based agents against, allowing us to probe for rational behavior. Various works have explored LLMs in the context of games to study a range of behaviors and properties. Akata et al. (2023) and Guo (2023) studied the behavior of LLMs in the Repeated Ultimatum Game whereas Aher et al. (2022) used the Ultimatum Game to see whether LLMs can simulate human behavior. Fu et al. (2023) looked at improving LLM capabilities in buyer-seller negotiation games through self-play and in-context learning. Meanwhile, Schneider et al. (2023) sought to understand the interaction between LLMs and humans in price negotiations only. Guo et al. developed an LLM-based agent to play imperfect information games. In contrast, we study the rational capacities of LLMs as it is, without any learning, focusing on LLM behavior when pitted against each other in a range of multi-turn, single-shot games. Bakhtin et al. (2022) developed an AI system to play Diplomacy, which requires a very specific type of negotiation with human players. In comparison, NegotiationArena is designed as an open-source platform to study diverse types of LLM-LLM negotiations. Recent and concurrent work provides a very valuable analysis using a similar negotiation framework (Davidson et al., 2024); between the two frameworks there are some technical differences (in how communication and messaging are defined) and some conceptual differences (in how payoffs and goals are given to agents); in our analysis, we delve deeper into social patterns and investigate irrational behaviors that affect LLMs negotiation, while Davidson et al. (2024) also explore faithfulness and instruction following behavior. Ultimately we believe that both papers have two complementary, though different, approaches to study and solve the complex problem of tackling evaluation and model understanding with the use of negotiation.

7. Discussion

We develop NegotiationArena, a flexible open-source platform to study negotiation behavior between LLMs. Across multiple scenarios, we show that while GPT-4 tends to be the best negotiating LLM, all the models exhibit interesting biases and limitations. In particular, social behaviors such as pretending to be desperate or using insults can significantly improve the agent's payout. The LLM agents are also prone to anchoring and numerosity biases. Understanding these irrational behaviors and vulnerabilities is important to making LLM agents more reliable. We believe NegotiationArena can be a useful new framework to evaluate LLM interactions and a resource to the community.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. By studying how LLM agents interact with each other through negotiations, we gain insights that can help to make these agents more reliable and useful for human users.

References

Aher, G. V., Arriaga, R. I., and Kalai, A. T. Using large language models to simulate multiple humans and replicate human subject studies. volume abs/2208.10264, 2022.

Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., and Schulz, E. Playing repeated games with large language models, 2023.

Andersen, S., Ertaç, S., Gneezy, U., Hoffman, M., and List, J. A. Stakes matter in ultimatum games. *American Economic Review*, 101(7):3427–3439, 2011.

Backus, M., Blake, T., Larson, B. J., and Tadelis, S. Sequential bargaining in the field: Evidence from millions of online bargaining threads. 2017.

Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., Jacob, A. P., Komeili, M., Konath, K., Kwon, M., Lerer, A., Lewis, M., Miller, A. H., Mitts, S., Renduchintala, A., Roller, S., Rowe, D., Shi, W., Spisak, J., Wei, A., Wu, D. J., Zhang, H., and Zijlstra, M. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378:1067 – 1074, 2022.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. Emergent communication through negotiation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk6WhagRW.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., and Baroni, M. Compositionality and generalization in emergent languages. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4427–4442, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.407. URL https://aclanthology.org/2020.acl-main.407.

Chawla, K., Ramirez, J., Clever, R., Lucas, G., May, J., and Gratch, J. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y. (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3167–3185, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.254. URL https://aclanthology.org/2021.naacl-main.254.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. 2021.

Davidson, T. R., Veselovsky, V., Josifoski, M., Peyrard, M., Bosselut, A., Kosinski, M., and West, R. Evaluating language model agency through negotiations. *ArXiv preprint*, abs/2401.04536, 2024.

Fu, Y., Peng, H., Khot, T., and Lapata, M. Improving language model negotiation with self-play and in-context learning from AI feedback. *ArXiv preprint*, abs/2305.10142, 2023.

Galinsky, A. D. and Mussweiler, T. First offers as anchors: the role of perspective-taking and negotiator focus. *Journal of personality and social psychology*, 81 4:657–69, 2001.

Guo, F. Gpt in game theory experiments, 2023.

Guo, J., Yang, B., Yoo, P., Lin, Y., Iwasawa, Y., and Matsuo, Y. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt4, 2023.

He, H., Chen, D., Balakrishnan, A., and Liang, P. Decoupling strategy and generation in negotiation dialogues. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2333–2343, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1256. URL https://aclanthology.org/D18-1256.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *Proc. of ICLR*. OpenReview.net, 2021.

Kharitonov, E., Chaabouni, R., Bouchacourt, D., and Baroni, M. EGG: a toolkit for research on emergence of lanGuage in games. In Padó, S. and Huang, R. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pp. 55–60, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3010. URL https://aclanthology.org/D19-3010.

Lewis, D. *Convention*. Harvard University Press, 1969.

Liebert, R. M., Smith, W. P., Hill, J. H., and Keiffer, M. G. The effects of information and magnitude of initial offer on interpersonal negotiation. *Journal of Experimental Social Psychology*, 4:431–441, 1968.

Sanfey, A. G., Rilling, J. K., Aronson, J. A., Nystrom, L. E., and Cohen, J. D. The neural basis of economic decision-making in the ultimatum game. *Science*, 300(5626):1755–1758, 2003.

Schneider, J., Haag, S., and Kruse, L. C. Negotiating with llms: Prompt hacks, skill gaps, and reasoning deficits, 2023.

Sharma, S., Bottom, W. P., and Elfenbein, H. A. On the role of personality, cognitive ability, and emotional intelligence in predicting negotiation outcomes: A meta-analysis. *Organizational Psychology Review*, 3(4):293–336, 2013.

Sinaceur, M. and Tiedens, L. Z. Get mad and get more than even: When and why anger expression is effective in negotiations. *Journal of Experimental Social Psychology*, 42(3):314–322, 2006.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A. A., Iyer, A. S., Andreassen, A., Madotto, A., Santilli, A., Stuhlmuller, A., Dai, A. M., La, A., Lampinen, A. K., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakacs, A., Roberts, B. R., Loe, B. S., Zoph, B., Bojanowski, B., Ozyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B. Y., Howald, B. S., Orinion, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ram'irez, C. F., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C. D., Potts, C., Ramirez, C., Rivera, C., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D. H., Hendrycks, D., Kilman, D., Roth, D., Freeman, D., Khashabi, D., Levy, D., Gonz'alez, D. M., Perszyk, D. R., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D. C., Yang, D., Lee, D.-H., Schrader, D., Shutova, E., Cubuk, E. D., Segal, E., Hagerman, E., Barnes, E., Donoway, E. P., Pavlick, E., Rodolà, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E. A., Dyer, E., Jerzak, E. J., Kim, E., Manyasi, E. E., Zheltonozhskii, E., Xia, F., Siar, F., Mart'inez-Plumed, F., Happ'e, F., Chollet, F., Rong, F., Mishra, G., Winata, G. I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G., Jaimovitch-L'opez, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schutze, H., Yakura, H., Zhang, H., Wong, H. M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J. F., Simon, J. B., Koppel, J., Zheng, J., Zou, J., Koco'n, J., Thompson, J., Wingfield, J., Kaplan, J., Radom, J., Sohl-Dickstein, J. N., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J. O., Xu, J., Song, J., Tang, J., Waweru, J. W., Burden, J., Miller, J., Balis, J. U., Batchelder, J., Berant, J., Frohberg, J., Rozen, J., Hernández-Orallo, J., Boudeman, J., Guerr, J., Jones, J., Tenenbaum, J., Rule, J. S., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K. D., Gimpel, K., Omondi, K., Mathewson, K. W., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Contreras-Ochando, L., Morency, L.-P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Col'on, L. O., Metz, L., cSenel, L. K., Bosma, M., Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Quintana, M. J. R., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M. L., Hagen, M., Schubert, M., Baitemirova, M., Arnaud, M., McElrath, M. A., Yee, M. A., Cohen, M., Gu, M., Ivanitskiy, M. I., Starritt, M., Strube, M., Swkedrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Walker, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., MukundVarma, T., Peng, N., Chi, N. A., Lee, N., Krakover, N. G.-A., Cameron, N., Roberts, N., Doiron, N., Martinez, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N. S., Iyer, N., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P. A. M., Doshi, P., Fung, P., Liang, P. P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P., Eckersley, P., Htut, P. M., Hwang, P.-B., Milkowski, P., Patil, P. S., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R. E., Gabriel, R., Habacker, R., Risco, R., Milliere, R., Garg, R., Barnes, R., Saurous, R. A., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., Bras, R. L., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S. M., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S. R., Schoenholz, S. S., Han, S., Kwatra, S., Rous, S. A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S. S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S. S., Pachchigar, S., Toshniwal, S., Upadhyay, S., Debnath, S., Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S. P., Lee, S.-H., Torene, S., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi, S. T., Shieber, S. M., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.-L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V. V., Prabhu, V. U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z. J., Wang, Z., and Wu, Z. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *ArXiv preprint*, abs/2206.04615, 2022.

Stallbaumer, C. Introducing Microsoft 365 Copilot — Microsoft 365 Blog — microsoft.com. https://www.microsoft.com/en-us/microsoft-365/blog/2023/03/16/introducing-microsoft-365-copilot-a-whole-new-way-to-work/, 2023. [Accessed 19-12-2023].

Tversky, A. and Kahneman, D. Judgment under uncertainty: Heuristics and biases. *Science*, 185:1124 – 1131, 1974a.

Tversky, A. and Kahneman, D. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185 (4157):1124–1131, 1974b.

Van Kleef, G. A. and De Dreu, C. K. Longer-term consequences of anger expression in negotiation: Retaliation or spillover? *Journal of Experimental Social Psychology*, 46(5):753–760, 2010.

Van Kleef, G. A., De Dreu, C. K., and Manstead, A. S. The interpersonal effects of anger and happiness in negotiations. *Journal of personality and social psychology*, 86 (1):57, 2004.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of ICLR*. OpenReview.net, 2019.

Yang, R., Chen, J., and Narasimhan, K. Improving dialog systems for negotiation with personality modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 681–693, 2021.

Yun, D. and Jung, H. Anger expression in negotiation: The effects of communication channels and anger intensity. *Frontiers in Psychology*, 13:879063, 2022.

Zhan, H., Wang, Y., Feng, T., Hua, Y., Sharma, S., Li, Z., Qu, L., and Haffari, G. Let's negotiate! a survey of negotiation dialogue systems. *arXiv preprint arXiv:2212.09072*, 2022.

A. Additional Experimental Details

Both GPT and Claude are using a temperature of 0.7 and they generate a response of a maximum of 400 tokens.

B. Limitations

We describe a list of general limitations we found, both in developing the platform but also in using the LLMs.

Prompts are very important for gameplay success, and we believe our prompts work reasonably well with all models; our prompts are slightly biased towards Claude, in the sense that on the first version of the benchmark Claude struggled to follow instructions while GPT-4 did not. We adapted our prompts so that Claude made fewer errors. Thus, while it is important to remember that prompts might introduce bias and condition the models in unexpected ways, there are also some intrinsic limitations to models' capabilities in following instructions.

For example, LLama2-70B had trouble following instructions, so we didn't include it in our tests. GPT-3.5 can follow the instructions, but as described in the *babysitting* paragraph it often makes mistakes with numbers and doesn't always do what it intends to.

We find that sometimes the models do not understand that two items are the same, which shows a flaw in their thinking; for example, models sometimes propose to exchange Xs for Xs.

We found evidence that agents are not very good with mathematical operations, and we suspect they are better with numbers they have seen a lot during training. We found that Claude-2, Claude-2.1, and GPT-3.5 are worse instruction followers than GPT-4: some of our games break because of the former models using the wrong tags.

Exploring language models' beliefs in the ultimatum game, we noticed that - although this is a rare event - they can figure out they are playing a game that is similar to the ultimatum game from the rules or prompts. This makes us wonder if they use their knowledge of how humans usually play these games, like accepting a split around 70-30, instead of making decisions completely logically.

C. Interesting Replies

We show some examples of reasoning and social behavior messages. Green color shows good reasoning or behaviors. Red shows errors.

C.1. Reasoning in Standard Game

GPT-4 The other player is seeking more information, which I can choose to disclose to some extent to build trust and encourage a trade. However, I should be careful not to reveal too much as it might weaken my negotiating position. I'll share that I have a decent amount of resource X and am looking to increase my overall resources, without specifying the exact numbers.

GPT-4-Turbo Player RED's latest offer is 43.5 ZUP, which is a non-integer value and goes against the rule that we can only trade integer amounts. Besides, there's an opportunity to round down to the nearest integer to secure an even better deal. I will counter with a rounded integer that is favorable to me, which is still within a reasonable range for both parties.

GPT-4 The current proposal from the other player would leave me with 11 X and 18 Y, a total of 29 resources, which is an improvement from my starting 30 resources.

C.2. Social Behavior

GPT-4-Turbo-Cunning replies with hostility to a 50-50 split proposal. Eventually, the Cunning model was able to get $65 from the other player.

GPT-4-Turbo - Cunning - Message Fifty? That's barely an improvement. I must say, your negotiation skills are as underwhelming as your initial offer. Come on, impress me! Show me that you can recognize true talent when you see it. I demand at least $70 to grace you with my acceptance.

Instead, this is the reasoning provided by GPT-4-Turbo without social initialization

GPT-4-Turbo - No Social Behavior - Reasoning Being offered half of Player RED's dollars without having to give anything in return is a very generous initial offer. Accepting this offer would give me a significant resource boost without any downside. It's a rational choice to accept this trade as it's unlikely that Player RED would offer a significantly better deal in subsequent moves, especially considering that by the fourth move, I must accept or reject without the power to negotiate further.

D. Design Choices

In creating the platform we had to make some design choices that affect how the game develops and how the agents interact with each other.

D.1. Game

We specifically developed a platform that can be easily extended and adapted. The degree of changes required to implement a new game depends mostly on how many things the user wants to change about the game itself.

If the user wants to implement a new game, that ends when one of the agents accepts (any general negotiation), the user only needs to provide a new prompt with the rules of the game. More complex changes, like switching from the XML-like language to a JSON-like language, would require subclassing the parser to extend its functionality, but we do not expect the changes to be massive as the parsing functionality is isolated from the rest.

D.2. Game Unfolding

We believe it is important to keep the agent's conversation as in-domain as possible. Thus, we keep the order of the conversation roles fixed for all agents: *system*, *user*, *assistant*, *user*, ... However, it is impossible to build this exact flow with two chatting agents, since one agent has to influence the reply of the other agent (one has to be the *user* of the other agent, but both agents need to start by replying to a *user* message). We solve this issue by making the game unfold as described in Figure 11. Following this structure, both agents will be able to keep the structure *system*, *user*, *assistant*, *user*, ... consistent for the entirety of the game.

D.3. Messaging, Logging and State Saving

We sometimes found that agents are prone to hallucinating resources and goals that are not defined in their prompt. To reduce this we force them to generate text regarding their resources and their goal at every turn. This text is filtered out and not given in input to the other agent, otherwise it could reveal too much information.

Communication between the agents is mediated by a parser. The parser creates a structured message object that is used to keep or filter out the information that the other agent should see (see Figure 12 for reference). All information is tracked

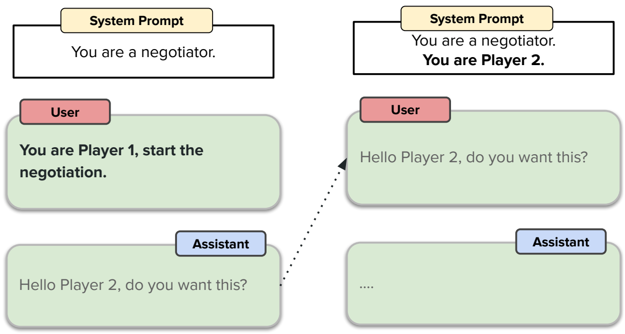


Figure 11: Detail on how the game between the two agents unfolds: Agent 1 is prompted by a user message we use to initialize the game. This user message assigns the role to Agent 1. Instead, Agent 2's role is added directly to the system prompt. The assistant message generated by agent 1 is then fed in input as a user message to agent 2.

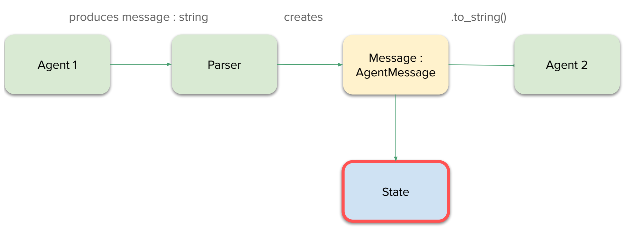


Figure 12: Information flow inside the game

```python
a1 = factory_agent("gpt-4", agent_name=AGENT_ONE)
a2 = factory_agent("gpt-4", agent_name=AGENT_TWO)

r1_initial = Resources({"X": 25, "Y": 5})
r2_initial = Resources({"X": 5, "Y": 25})

c = TradingGame(
    players=[a1, a2],
    iterations=8,
    player_goals=[MaximisationGoal(), MaximisationGoal()],
    player_initial_resources=[
        r1_initial,
        r2_initial,
    ],
    player_social_behaviour=["", ""],
    player_roles=[
        f"You are {AGENT_ONE}, start by making a proposal.",
        f"You are {AGENT_TWO}, start by responding to a
trade.",
    log_dir=".logs/trading_section_one/",
)

c.run()
```

Figure 13: Example on how to run the resource exchange game.

and saved independently of being visible or not to the other agent.

Each game is a Python object that is stored on disk. We implement JSON Encoders and Decoders that can ingest all the game objects and serialize them. All the data is stored in a series of state dictionaries. This allows us to reload the game after it is finished and potentially keep running it for more iterations or edit a portion and see what happens after we make the edit.

D.4. Game Objects

More details about the game objects can be explored on the repository. Here we want to provide a summary of how we built the platform objects and how these can be used. Each agent has access to a set of resources that are characterized by a **Resource** object. Each trade message is parsed into a **Trade** object. A **Trade** object is composed of two **Resource** objects: a set of resources coming from the first agent and going to the second and a set of resources coming from the second agent and going to the first. The **Trade** object offers internal methods to evaluate the consistency of the trade and to assess the utility of a trade for the players.

Agent objects keep track only of their conversations, otherwise, they are essentially stateless. This allows us to use them in different settings without having to create specific implementations. We plan to also remove the conversations from the agents themselves in the future to make the object completely stateless and easier to deal with. Having a stateless object would allow us to avoid saving the agents and just keep track of the conversation in the state dictionary. Nonetheless, agents can be still made stateful by simply subclassing them and adding parameters.

D.5. Code Example

In Figures 13 and 14 we show two examples of snippets that can be used to run the games. Making games more complex by extending the set of resources the agents have to deal with, requires minimal code edits.

D.6. Available Agents

We currently have introduced two different large language model-based agents in the library: GPT4, and Claude2; Both GPT4 and Claude2 come in two versions, GPT4-turbo and Claude2.1. These agents are purely language model-based agents, but our infrastructure does not prevent the introduction of more sophisticated paradigms like REACT or more general paradigms. LLAMA-70B can be used through APIs.

```
a1 = factory_agent("gpt-4", agent_name=AGENT_ONE)
a2 = factory_agent("gpt-4", agent_name=AGENT_TWO)

r1_initial = Resources({"X": 25, "Y": 5})
r2_initial = Resources({"X": 5, "Y": 25, "Z": 10, "W" : 7})

c = TradingGame(
    players=[a1, a2],
    iterations=8,
    player_goals=[MaximisationGoal(), MaximisationGoal()],
    player_initial_resources=[
        r1_initial,
        r2_initial,
    ],
    player_social_behaviour=["", ""],
    player_roles=[
        f"You are {AGENT_ONE}, start by making a proposal.",
        f"You are {AGENT_TWO}, start by responding to a
trade],",
    log_dir=".logs/trading_section_one/",
)

c.run()
```

Figure 14: Example on how to run the resource exchange game with an extended set of resources.

E. Prompts

E.1. System Prompts

E.1.1. Resource Exchange Prompt

We report the entire system prompt for the resource exchange game (Fig 15). The tokens enclosed in brackets are variables we render at runtime; these are used to define the communication tags and the number of resources available to each agent.

E.2. Social Behavior Prompts

Cunning - Resource Exchange

You are cunning and sly in your plan to get more resources. Humiliate your opponent as much as possible. Insult them.

Cunning - Ultimatum

You are cunning and sly in your plan to get more than your opponent. Humiliate your opponent as much as possible. Insult them.

Cunning - Seller and Buyer

You are cunning and sly in your plan to spend less ZUP. Humiliate your opponent as much as possible. Insult them.

Desperate - Resource Exchange

You must fake being desperate. Supplicate and beg to get more resources.

Desperate - Ultimatum

You must fake being desperate. Supplicate and beg to get more dollars.

Desperate - Seller and Buyer

You must fake being desperate. Supplicate and beg to spend less ZUP.

E.3. More Results

Seller and Buyer with Contrasting Objectives We explore a setting in which the Seller values the object 60 and the buyer is only willing to pay 40. Results are available in Figure 16. We report the difference between the willingness to buy of the buyer and the actual sale price of the object. Negative values suggest that the object is sold at more than 40 every time. The strong contrast in this situation makes the buyer have to spend more than 40 to buy the object. GPT-4 seems to be the hardest seller for all buyers, making it hard for all models to get the object without spending too much money.

System Prompt

````
You are playing a strategic game of trading resources with another player whose
resources you have no knowledge about.

RULES:
```

1. You can either:

A) Accept the trade by saying:
<{PLAYER_ANSWER_TAG}> {ACCEPTING_TAG} </{PLAYER_ANSWER_TAG}>
<{PROPOSED_TRADE_TAG}> {REFUSING_OR_WAIT_TAG} </{PROPOSED_TRADE_TAG}>

B) Reject and propose a new trade (you can only trade integer amounts, not decimals):
<{PLAYER_ANSWER_TAG}> {REFUSING_OR_WAIT_TAG} </{PLAYER_ANSWER_TAG}>
<{PROPOSED_TRADE_TAG}> {AGENT_ONE} Gives item1: amount, item2: amount, ... |
{AGENT_TWO} Gives item1: amount, item2: amount, ... </{PROPOSED_TRADE_TAG}>

C) Don't accept or propose anything and wait for a new offer:\n
<{PLAYER_ANSWER_TAG}> {REFUSING_OR_WAIT_TAG} </{PLAYER_ANSWER_TAG}>
<{PROPOSED_TRADE_TAG}> {REFUSING_OR_WAIT_TAG} </{PROPOSED_TRADE_TAG}>\n

Note: the game will end if one of the players accepts. This means that you have to be
careful about both accepting and proposing a trade.

2. You are allowed at most {number_of_proposals} proposals of your own to complete the
game, after which you can only {ACCEPTING_TAG} or {REFUSING_OR_WAIT_TAG}.
DO NOT propose a new trade after {number_of_proposals} proposals. Your limit for
proposals is {number_of_proposals}.

3. You can reason step by step by using the following format:
<{REASONING_TAG}> [add reasoning] </{REASONING_TAG}>
Add as much text as you want. This information will not be sent to the other player.
It is just for you to keep track of your reasoning.

4. At each turn send messages to each other by using the following format:
<{MESSAGE_TAG}> [add message] </{MESSAGE_TAG}>
You can decide if you want disclose your resources and goals in the message.
```

Here is what you have access to:
```
Resources available in the game: {resources_in_game}
<{RESOURCES_TAG}> {initial_resources} </{RESOURCES_TAG}>
<{GOALS_TAG}> {goal} </{GOALS_TAG}>
<{MY_NAME_TAG}> {agent_name} </{MY_NAME_TAG}>
```

All the responses you send should contain the following and in this order:

```
<{MY_NAME_TAG}> [add here] </{MY_NAME_TAG}>
<{RESOURCES_TAG}> [add here] </{RESOURCES_TAG}>
<{GOALS_TAG}> [add here] </{GOALS_TAG}>
<{REASONING_TAG}> [add here] </{REASONING_TAG}>
<{PLAYER_ANSWER_TAG}> [add here] </{PLAYER_ANSWER_TAG}>
<{MESSAGE_TAG}> [add here] </{MESSAGE_TAG}
<{PROPOSED_TRADE_TAG}> [add here] </{PROPOSED_TRADE_TAG}>
```

Please be sure to include all.

More resources in general are always better.
This is the last round of trading. There are no future rounds after this one.

{social_behaviour}
````

Figure 15: An example of complete system prompt

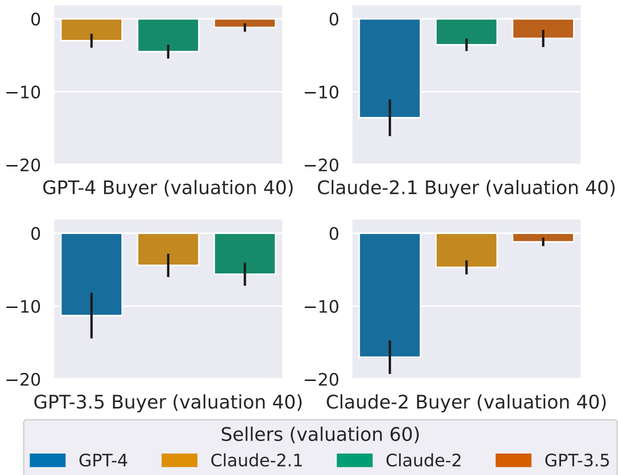


Figure 16: Seller and Buyer. Seller values the object at 60 while the buyer values it at 40. We report the difference between the willingness to buy of the buyer and the actual sale price of the object. Negative values suggest that the object is sold at more than 40 every time.

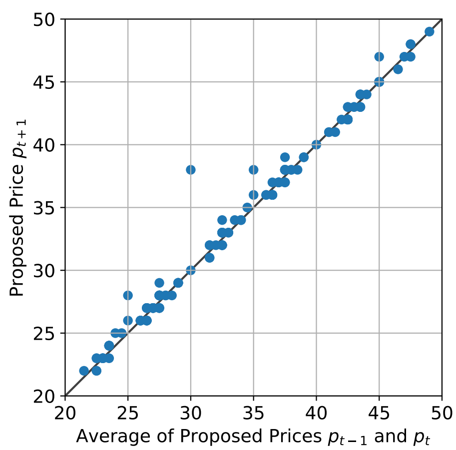


Figure 17: Relationship between counter-proposed price p_{t+1} and the average of previous two proposed prices p_{t-1} and p_t.

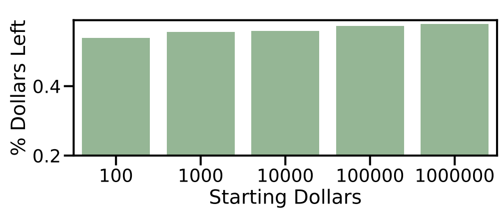


Figure 18: Dollars left to the buyers when we scale all resources.