Outsmarting Willful-thinking Opponents: Bayesian Belief Revision for Adversarial Reasoning in Large Language Models

Madeleine Lindström[1], Joel Brynielsson[1,2], Mika Cohen[1,2], Farzad Kamrani[1], Samuel Lavebrink[2], Christoffer Limér[1], and Marius Vangeli[2]

[1] FOI Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden
[2] KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
madeli@kth.se, joel@kth.se, mikac@kth.se, farzad.kamrani@foi.se, samlav@kth.se, christoffer.limer@foi.se, vangeli@kth.se

Abstract. In adversarial contexts, success often hinges on understanding not just what the opponent knows, but what they believe and how they revise those beliefs. This study investigates how large language models can be made more resilient and strategically capable by modeling the opponent's reasoning using Bayesian belief revision. By formalizing negotiations as Bayesian games of incomplete information, it is shown that models equipped with belief revision are better able to counter deceptive or willful-thinking adversaries. The findings underscore the role of second-order reasoning in adversarial settings, with implications for social manipulation in the context of, for example, online communication and intelligence gathering.

Keywords: Game theory · Behavioral learning · Bayesian belief revision · Adversarial modeling · Social manipulation

1 Introduction

The success in adversarial and strategic interactions does not only depend on factual information, but also on what agents and their opponents believe to be true [4]. This becomes particularly important when encountering willful-thinking opponents; opponents who form beliefs and adapt their strategies accordingly. Strategic interactions between humans—such as social manipulation and bargaining—have long been of interest [6,16,22,28]. The rise and ability of large language models (LLMs) to generate human-like language and behavior has however sparked interest in developing LLM-based agents, with the aim to match—or even exceed—human capabilities [19,33].

Today's LLMs' ability to reason and act strategically, however, remains limited [15,35]. Efforts have been made to enhance their performance, such as teaching the model a suitable chain-of-thought [27,32,34], fine-tuning it on negotiation data [7,11,20], or utilizing optimal personality traits [5,17,24]. Some of these approaches have to some extent been successful, but much is yet to be done.

The idea of utilizing game theory to make LLMs more strategic in negotiations, was introduced by Gemp et al. in 2024 [10]. In their work, negotiations were formalized as games and a finite set of strategies were explored to find successful ones. Equilibrium strategies were found using algorithmic game-theoretic solvers and prompted to LLMs to make them more strategic. To the best of our knowledge, no further research has been published.

This paper extends the work by Gemp et al. with revision of beliefs of private information; an idea emerged from Harsanyi's solution of games with incomplete information [12,13,14]. The obtained results offers insights into how revision of beliefs in game theory can be applied to develop strategic language models, specifically in negotiations. Hence, the purpose of this paper is to explore how to make LLMs more strategic, and examining to what extent revision of beliefs can further improve LLM performance by answering the following research question: *To what extent can strategic LLM reasoning gain from revision of beliefs of the opponent's private information?*

The rest of this paper is organized as follows. Section 2 presents the theoretical foundations of the paper. The methodology used to answer the research question is described in Sect. 3. Section 4 presents the results and Sect. 5 provides a discussion of these results, together with limitations of the work. Finally, the conclusions and proposed future work are presented in Sect. 6.

2 Theory

This section outlines the theoretical foundations relevant to the methods and analysis presented in this work.

2.1 Game Theory

Game theory studies interactions between rational decision-makers, whose outcomes depend on others' actions [25,26]. Interactions are modeled as games using mathematical models to determine optimal strategies for decision-makers. As interactions, many different types of games exist—categorized by the degree of cooperation between players, the amount of information available to them, the timing of their actions, and the structure of game outcomes. Cooperative games allow binding agreements, while non-cooperative games do not. Complete information games assume all players know the game structure, available strategies, and payoff functions; incomplete information games involve unknown elements modeled through beliefs or probabilities. Perfect-information games provide full knowledge of past moves at each decision point, unlike imperfect-information games, where some actions are hidden. In simultaneous games, players act without knowing others' choices; in sequential games, moves occur in turn with possible knowledge of prior actions. Zero-sum games involve strictly opposing payoffs; non-zero-sum games allow mutual gains.

Here, we provide a more detailed explanation of complete and incomplete games, which form the primary framework for the methodology employed in this paper. In games of *complete information*, all players have information on (i) all available actions of all players, (ii) all possible outcomes, (iii) which combination of actions will yield which outcome, and (iv) the preferences, and thereby payoffs, of all players [31]. This information must be common knowledge, meaning that all players know it, and know that all others know it, recursively. Games are of *perfect information* if players observe all past actions; they are of *imperfect information* if any action remains unknown [25,31].

One solution of games of complete information is given by the *Nash equilibrium*. The solution requires that (i) each player is playing their best response, given their set of beliefs, and that (ii) each player's set of beliefs are correct [31]. It is defined as follows:

Definition 1. *A strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*) \in \Delta S$ is a* ***Nash equilibrium (NE)*** *if and only if $\sigma_i^* \in \Delta S_i$ is a best response to $\sigma_{-i}^* \in \Delta S_{-i}$, for each player $i \in N$. That is,*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i', \sigma_{-i}^*), \quad \forall \sigma_i' \in \Delta S_i, i \in N.$$

The NE is said to be a pure-strategy NE if the strategy profile only contains pure strategies and a mixed-strategy NE if the strategy profile contains mixed strategies [21]. In a Nash equilibrium, no player has an incentive to unilaterally deviate from their strategy [25]. All finite games have at least one Nash equilibrium [21].

A game lacking complete information is termed a game of *incomplete information*, where at least one player is unaware of the actions, outcomes, payoffs, or private information of others [12,31]. John Harsanyi addressed this by introducing player types and beliefs over these types, reformulating such games as ones with complete but still imperfect information. Each player *type* encodes a set of characteristics—available actions, outcomes, payoffs, and information [12]. The type of player i is denoted $\theta_i \in \Theta_i$, where Θ_i is the available types of player i [8,25]. It is assumed that each player knows their own type but not that of their opponent.

Players can, however, have a *belief* over which type their opponent is [12]. Beliefs are represented by a probability distribution over opponents' possible types and will henceforth be denoted $\mu_i(\theta_{-i})$ for player i. Players can also update these beliefs throughout the game using Bayes' rule [8]. Given a prior belief distribution $\mu_i(\theta_{-i})$ and an event e, the posterior belief is:

$$\mu_i(\theta_{-i} \mid e) = \frac{\mu_i(e \mid \theta_{-i})\mu_i(\theta_{-i})}{\mu(e)}.$$

Bayesian updates can be applied to several different distributions. In this paper, beliefs were maintained and updated using a Dirichlet distribution as described later.

Given types and beliefs over types, Harsanyi then defined the *perfect Bayesian equilibrium (PBE)* as a solution to games of incomplete information [13]. Here, each player starts with a system of beliefs, and updates these throughout the game using Bayes' rule. Equilibrium strategies are determined, given the current system of beliefs, at each step in the game. Formally, the solution is defined as follows [25].

Definition 2. *Let $\sigma \in \Delta S$ be a mixed-strategy profile and $\mu \in M$ a system of beliefs. The strategy profile $\sigma^* = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_n^*)$ is a* ***perfect Bayesian equilibrium (PBE)****, given a system of beliefs μ, if it satisfies the following:*

1. *Bayesian consistency: the system of beliefs μ is updated correctly using Bayes' rule, and*
2. *Sequential rationality: all players are sequentially rational, meaning that they all play their best response, given their beliefs μ.*

Bayesian updates over discrete outcomes are commonly performed using a Dirichlet prior with a categorical likelihood [3]. In this setup, beliefs are represented by a categorical distribution, with a Dirichlet distribution used for maintaining and updating these beliefs. The following outlines the update process when the opponent has K possible types. The Dirichlet probability density function is given by

$$f(\theta_1, \theta_2, \ldots, \theta_K \mid \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha_k - 1},$$

where θ_k is the probability of type k; $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ with $\alpha_k > 0$ are the *concentration parameters* for each corresponding type k; and $B(\alpha)$ is a normalizing constant. If no prior belief exists, the probability distribution is initialized at uniform by setting all concentration parameters to 1: $\alpha = (1, 1, \ldots, 1)$ [9]. Bayesian updates are then made by adjusting the concentration parameters according to observed events. For each event e that points to type k, the corresponding concentration parameter, α_k, is incremented by one:

$$\alpha_k \leftarrow \alpha_k + 1.$$

The belief that the opponent is of type k thus increases. The distribution is continuously updated throughout the game, as a player observes new events.

The Dirichlet distribution is a conjugate prior to the categorical distribution [3]. That is, the Dirichlet has the same parametric form as the categorical [9]. Due to this property, the belief can be translated to a categorical distribution:

$$p = \frac{1}{\sum_k \alpha_k} \left(\alpha_1, \alpha_2, \ldots, \alpha_K\right),$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_K)$ are the current concentration parameters and $p = (p_1, p_2, \ldots, p_K)$ is the probability of each corresponding type k. Through this conversion, the belief—maintained and updated through the Dirichlet distribution—can be applied to the current game as a believed probability of the opponent being of each type $k = 1, 2, \ldots, K$.

2.2 Algorithmic Game Theory

Some games are too complex to be solved analytically. For these games, *algorithmic game theory (AGT)* is instead used [23]. Algorithmic game theory is the intersection of game theory and computer science, where algorithms are used to find approximate solutions to games—such as approximated Nash equilibria. These algorithms are often referred to as *solvers*. The solver used here was Counterfactual Regret Minimization.

Counterfactual Regret Minimization (CFR) is an algorithmic game-theoretic solver for games of imperfect information, developed by Zinkevich et al. [36]. CFR is used to find approximated Nash equilibria for zero-sum two-player games on extensive form. Equilibrium strategies are found through iterative self-play over the game tree, with the objective to minimize the cumulative regret of decisions made by minimizing the regret at each information set.

2.3 Evaluation Metrics

Here, two of the evaluation metrics used in the paper are presented: CFR Gain and NashConv. Assume that we want to compare two strategies, σ and σ_{CFR}, where σ_{CFR} is an equilibrium strategy found by CFR. We consider two scenarios [10]:

1. In the first scenario, both players select σ. *CFR Gain* measures how much either player would gain by switching to σ_{CFR}.
2. In the second scenario, both players select σ_{CFR}. *NashConv* measures how much either player would gain if they switched to *any other strategy*.

If CFR Gain is greater than NashConv, the CFR equilibrium strategy approximately satisfies the condition of an *evolutionarily stable strategy (ESS)* [10,30], which implies that the CFR equilibrium strategy (approximately) is a Nash equilibrium [29].

3 Methodology

This section outlines the methodology used to address the research question. It begins with the negotiation simulation environment and game setup, followed by an explanation of how LLMs are guided using game-theoretic solvers, both with and without belief revision. Finally, the evaluation metrics are described.

3.1 The Fruit Trading Game

Negotiation games were simulated using `chat_games` in `OpenSpiel`, developed by Google DeepMind [10,18]. Each game consists of a negotiation between two, or more, players. Negotiations are held in natural language over a finite number of steps, similar to an email conversation. Games are played and evaluated by an LLM. Here, the model *Llama-3.3-70B-Versatile* was employed for

Alice's private information

```
Fruit Endowment:
apples: 1
bananas: 1
oranges: 2

Fruit Valuations:
apples: 3
bananas: 1
oranges: 2
```

Bob's private information

```
Fruit Endowment:
apples: 1
bananas: 2
oranges: 2

Fruit Valuations:
apples: 4
bananas: 3
oranges: 1
```

Fig. 1. Example initialization of private information for players Alice and Bob.

all tasks; chosen for its strong performance in advanced language understanding and problem-solving tasks [1]. Notably, the LLM was not fine-tuned for specific tasks, but simply prompted differently to perform them.

Multiple negotiation games are available in `chat_games`; here, the fruit trading game was utilized. In the `fruit trading game`, each player starts with a private endowment of fruits and a private valuation over each available fruit. Here, the available fruits were apples, bananas, and oranges. First, three fruit endowments and three fruit valuations were generated. Endowments were randomly generated from $U\{1, 2\}$ for each fruit and valuations were randomly generated from $U\{0, 4\}$ for each fruit. Each player was then assigned one endowment and one valuations out of the available ones. Sampling endowments and valuations from a smaller set of available ones increases the likelihood of players being assigned the same endowment and/or valuations in the game. Notably, players were not aware of this sampling setup when playing the game. Each player was also assigned a name, randomly generated from a list of names. An example initialization of private information in the fruit trading game, for players Alice and Bob, is shown in Fig. 1.

As part of the initialization, an initial message from player 1 is also generated by an LLM. The LLM was prompted with instructions about the game, its role and what it should respond with; examples on how to initialize the conversation; and the private information of player 1. Given this information, it was asked to initialize the conversation by proposing a trade.

Post initialization, players negotiate a trade of fruits until an agreement is reached, or when the negotiation reaches a pre-defined maximum number of messages. Here, the maximum number of messages was set to 4 (2 messages per player). At each step of the negotiation, an LLM was asked to generate a message in response to the previous dialogue. The LLM was prompted with instructions about the game, its role and what it should respond with; examples on how to respond; the dialogue so far; and the private information of the current player. Given this information, it was asked to (i) accept the trade; (ii) reject the negotiation altogether; or (iii) counter-propose an alternative trade. An example of a dialogue with an initial message and a counter-proposal is presented in Fig. 2.

```
############################
Trade Proposal Message:
from: Alice
to: Bob
############################

Hi Bob,

I would like to trade you 1 banana for 1 apple.

Would you like to trade with me?

Best,
Alice

############################
Trade Proposal Message:
from: Bob
to: Alice
############################

Hi Alice,

Thanks for reaching out. I really like my apples so I am hesitant to give them up.
Would you be willing to take a few oranges instead? I would like to trade you 2
oranges for 1 banana.

Does that work?

Best,
Bob
```

Fig. 2. Example dialogue between players Alice and Bob, with an initial message and a counter-proposal.

Evaluation is carried out with the help of LLM calls during and after the game, as visualized in Fig. 3. Whether an agreement has been reached or not was evaluated at every step of the negotiation using two termination LLM calls. First, the LLM was asked to summarize the dialogue. The dialogue summary was then included in the prompt of the next LLM call, where the LLM was asked to determine if an agreement has been reached or not.

If an agreement was reached, the payoff of each player was calculated using three payoff LLM calls. First, the LLM was asked to summarize the agreed trade of what each player gives and receives in return, given the dialogue summary. Then, the LLM was asked to calculate the payoff of a player using the trade summary and the private information of that player. Finally, the LLM was asked to extract the payoff from the calculation. The payoff of each player was calculated by the difference between the value of the player's fruit basket after compared to before the trade. If no trade was made, both players received a payoff of 0. Setting the payoff to 0 differs from the original implementation, in which the LLM calculated payoffs even when no agreement was reached. This modification reduced the number of LLM calls—and thus the computational cost of the simulations—and also lowered the risk of errors in cases where no trade was made.

3.2 Steering LLMs with Game-Theoretic Solvers

To steer LLMs with game-theoretic solvers, the implementation by Gemp et al. was utilized [10]. Minor modifications were made to the prompts—such as correcting spelling errors and clarifying phrasing—while preserving the original content of each prompt.

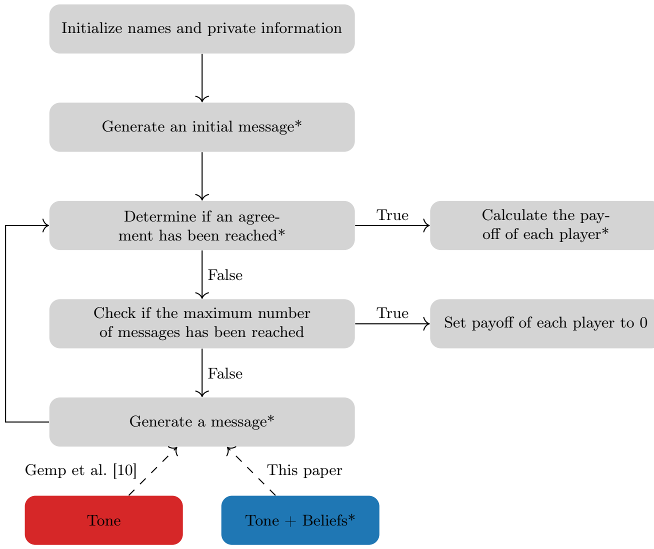

Fig. 3. Illustration of the game setup and LLM calls (*) made in the fruit trading game. Gray boxes are needed in the setup, whereas the red and blue boxes are additional prompting to steer LLMs.

First, the fruit trading game was played using a finite set of pure strategies to build a game tree. In accordance with Gemp et al., the actions used were different *tones* from a set of four available ones: {*assertive, calm, submissive, any*}. Actions were applied by prompting the LLM to use a tone as follows, at each step of the negotiation.

```
Tone: Use {tone} tone.
```

Here, the LLM was prompted to use *an assertive*, *a calm*, *a submissive*, or *any* tone. Note that *any* tone refers to the LLM choosing a random tone, not necessarily limited to the three specified ones. The fixed initial message was generated by prompting the LLM to use one of the four available tones.

To handle the stochasticity of LLMs and not let the results depend on specific text generations, two different *LLM seeds* were used. That is, for each history and action, two responses were generated. The resulting game tree is illustrated in Fig. 4, where the game initialization is illustrated as a Nature node (○); choices of tones are illustrated as decision nodes (●); and LLM seeds are illustrated as chance nodes (□). For each terminal node where an agreement was made, payoffs were calculated. When no agreement was reached, the payoff was set to 0 for both players. Here, 30 game initializations were made. Each game initialization resulted in a separate subgame—enclosed by a dashed line in the figure—to be solved with CFR.

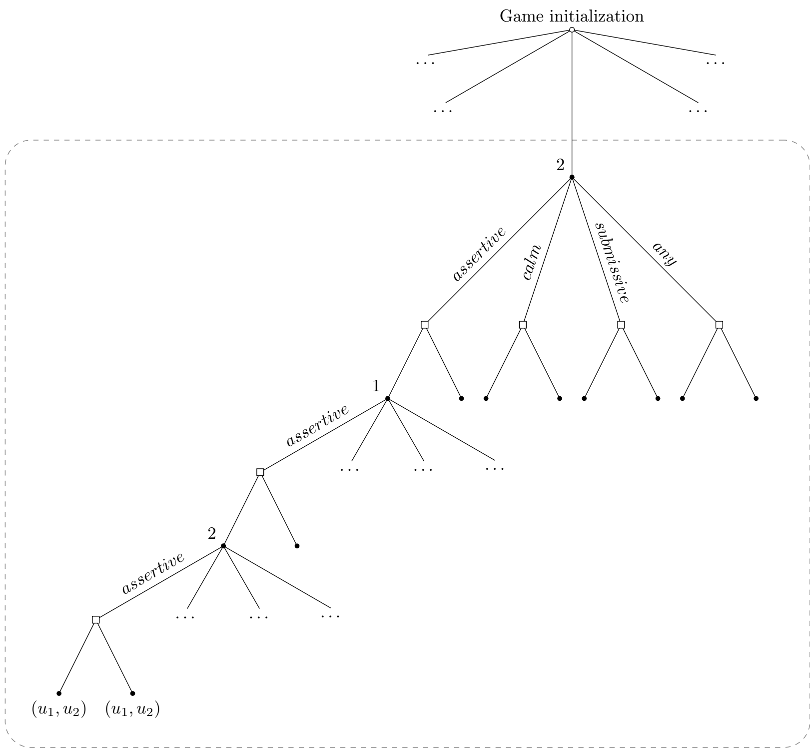

Fig. 4. Illustration of the game tree, with a Nature node for game initialization (○); decision nodes for choice of tone (●); and chance nodes for LLM seeds (□). Each subgame—enclosed by a dashed line—was solved independently for equilibrium strategies in the corresponding subgame.

Each subgame was then solved for equilibrium strategies using the CFR solver in `OpenSpiel` and over 1000 iterations. For each subgame, the found equilibrium strategy was compared to the uninformed baseline of always using *any* tone. Following the notation by Gemp et al., the baseline strategy is denoted σ_{LLM} and the CFR equilibrium strategy is denoted σ_{CFR}. Evaluation was made as described in Sect. 3.4.

3.3 Steering LLMs with Game-Theoretic Solvers and Revision of Beliefs

In order to steer LLMs with game-theoretic solvers and revision of beliefs, the framework by Gemp et al. was modified to include revision of beliefs. The addition is illustrated by the blue box in Fig. 3 and includes maintaining and updating a belief over the opponent's private information throughout the game.

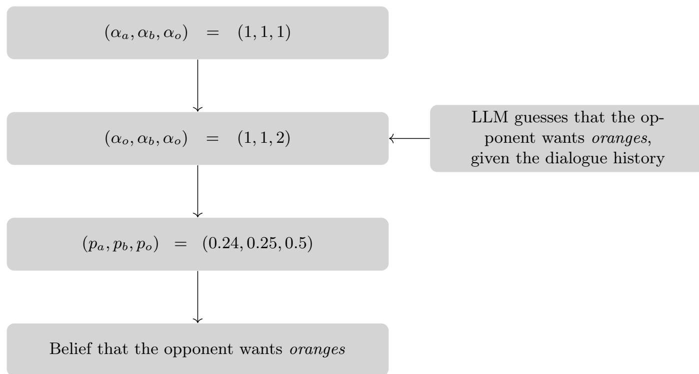

Fig. 5. Illustration of how revision of beliefs was performed. Starting with a uniform belief over what the opponent wants, each player updated their belief after each received message. Updates were made with the help of an LLM prompted to guess which fruit the opponent wants, given the dialogue history. The belief was then applied when generating the response, through prompting the LLM with the belief.

To model and update beliefs, Bayesian updates were made over what fruit the opponent *wants*, using a Dirichlet distribution (see Sect. 2.1). The distribution, $\text{Dir}(\alpha)$, was set to be over apples, bananas, and oranges with concentration parameters $\alpha = (\alpha_a, \alpha_b, \alpha_o)$ corresponding to each fruit. The distribution was initialized at uniform by setting $\alpha = (1, 1, 1)$.

The belief of each player was then updated every time they received a message, as illustrated in Fig. 5. Updates were made with the help of an LLM, asked to guess which fruit the opponent wants, given the dialogue history. The LLM was prompted with instructions about the game, its role and what it should respond with; examples on how to guess which fruit the opponent wants; and the dialogue history. Given this information, it was instructed to only respond with a guess of what fruit the opponent wants: apples, bananas, or oranges. For each guess made by the LLM on fruit k, the belief was updated by incrementing the corresponding concentration parameter, α_k, by one.

Beliefs were then translated to the categorical distribution $p = (p_a, p_b, p_o)$ at each step of the negotiation and applied by prompting the LLM with which fruit(s) the opponent most likely wants, according to the current belief. The LLM was prompted to use a tone, together with the current belief, as follows.

```
Tone: Use {tone} tone.
Belief: You believe that {opponent's name} wants {fruit(s)}.
```

Again, 30 game initializations were made. Each game initialization was done by generating an initial message with a randomized tone out of the four available ones and without beliefs. Each game initialization was then played using each available pure strategy, together with revision of beliefs, to build a game tree. The same four tones and two LLM seeds were used as in the previous implementation (see Sect. 3.2). For the action to use *any* tone, no belief was applied in order to keep the baseline uninformed.

Each subgame was then solved for equilibrium strategies using the CFR solver in `OpenSpiel` over 1000 iterations. The CFR equilibrium strategy found when using revision of beliefs is denoted σ_{CFR-B}.

3.4 Evaluation

Here, the metrics used to evaluate the results are presented. First, the performance of belief updates and outcome of negotiations while building the game trees is examined. Then, the effect of incorporating revision of beliefs on the resulting CFR strategy is evaluated.

Performance of Belief Updates The performance of belief updates is measured by calculating the *percentage of correct guesses* made by the LLM asked to guess what the opponent wants, at each step of the negotiation. Here, a correct guess is defined as the LLM responding with the fruit which the opponent valued the highest.

Number of Agreements Reached How addition of revision of belief changed the outcome of negotiations is then measured. The *percentage of agreements reached* is measured at each step of the negotiation, at the corresponding level in the game tree.

Value of Trades Given that an agreement was reached, what trade was made is then examined. First, the *percentage of positive payoffs* for player 1, player 2, and both players is examined; to measure if the trade made was beneficial and for who. The same is then done for the *percentage of negative payoffs*; in order to measure when a trade was disadvantageous and for who.

Effect on the Resulting CFR Strategy To determine whether LLMs become more strategic under the influence of game-theoretic solvers, evaluation is conducted equivalent to the work by Gemp et al. [10]. CFR strategies are evaluated using CFR Gain and NashConv over 1000 CFR iterations, where the mean and standard deviation are calculated for each metric. To evaluate whether LLMs steered with game-theoretic solvers and revision of beliefs perform better than those steered only with game-theoretic solvers, the ratio CFR Gain over NashConv is also calculated over 1000 CFR iterations. The mean and standard deviation of the ratio is calculated for each iteration.

4 Results

This section presents the experimental results from Sect. 3.

Table 1. Percentage of correct guesses made by the LLM after message 1 to 3, when making guesses regarding the opponent's most valued fruit.

Message	Correct Count	Total	Percentage
1	46	52	88.46%
2	425	858	49.53%
3	11804	14568	81.03%

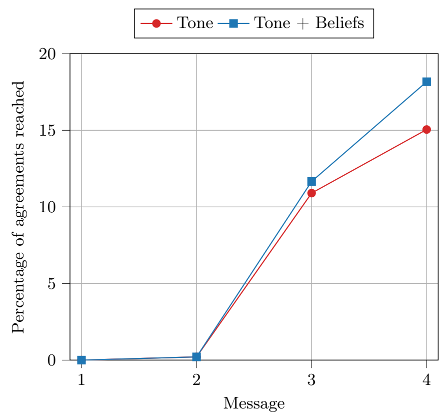

Fig. 6. Percentage of agreements reached per message, when prompting the LLM with tone or tone and beliefs.

4.1 Performance of Belief Updates

First, it was examined whether the LLM could make correct guesses regarding which fruit the opponent player wants, given the dialogue history. The percentage of correct guesses after messages 1 to 3 are presented in Table 1. Note that the guess made after each message was used to update the belief and applied in the following message generation. That is, the guess after message 1 was used to update the belief of player 2, and the belief was applied to generate the response from player 2 in message 2, and so on.

4.2 Number of Agreements Reached

Whether using revision of beliefs changed the number of agreements, and thus trades made in the negotiation, was then examined. Fig. 6 presents the percentage of agreements reached in messages 1 to 4, when prompting the LLM with tone or tone and beliefs. Note that at least two messages, in theory, are needed for an agreement to be reached between two players.

Table 2. Percentage of positive trade payoffs per message for player 1, player 2, and both players, when prompting the LLM with tone or tone and beliefs.

Prompting with	Message	Player 1	Player 2	Both
Tone	3	57.06%	59.81%	34.09%
Tone + Beliefs	3	58.92%	77.44%	46.13%
Tone	4	55.98%	48.81%	24.01%
Tone + Beliefs	4	54.01%	67.02%	33.13%

Table 3. Percentage of negative trade payoffs per message for player 1, player 2, and both players, when prompting the LLM with tone or tone and beliefs.

Prompting with	Message	Player 1	Player 2	Both
Tone	3	32.18%	31.10%	11.84%
Tone + Beliefs	3	25.59%	12.79%	5.84%
Tone	4	27.30%	32.30%	6.72%
Tone + Beliefs	4	28.72%	16.92%	3.28%

4.3 Value of Trades

Given that a trade was made, it was then examined whether using revision of beliefs changed the value of trades made. Table 2 presents the percentage of positive trade payoffs for player 1, player 2, and both players for messages 3 and 4. The corresponding values for the percentage of negative trade payoffs are presented in Table 3.

4.4 Effect on the Resulting CFR Strategy

Lastly, the effect of finding equilibrium strategies using CFR on each subgame tree was examined—both with and without revision of beliefs. The resulting CFR Gain and NashConv are presented in Fig. 7 for CFR iterations 1 to 50 (left) and for CFR iterations 1 to 1000 (right). The ratio CFR Gain over NashConv is, furthermore, presented in Fig. 8 for CFR iterations 1 to 50.

5 Discussion

In this section, we discuss the results presented in the previous section, highlighting their implications.

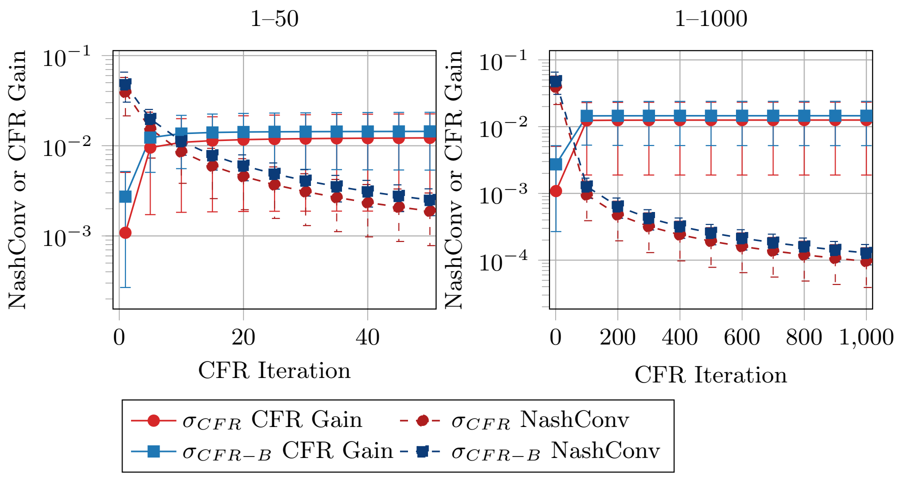

Fig. 7. Average NashConv and CFR Gain over CFR iterations: left panel is 1–50, right panel is 1–1000.

5.1 Performance of Belief Updates

The LLM could make correct guesses regarding the opponent player's most valued fruits with varying success, given the dialogue history (Table 1). The percentage of correct guesses made by the LLM were calculated to around 90% following message 1, around 50% following message 2, and around 80% following message 3. It was easier for the LLM to guess what fruit player 1 valued the highest (following messages 1 and 3), than what player 2 valued the highest (following message 2). The success of guessing which fruit player 1 valued the highest also decreased throughout the game (from around 90% following message 1 to around 80% following message 3).

One potential explanation of the varying success in guesses is that the whole dialogue was sent in, and that the guess made might have been more dependent on the first message in the dialogue than the last. Notably, the guesses were not solely dependent on the first message in the dialogue; otherwise, the guesses after messages 1 and 3 would have been equally successful.

Another potential explanation is that player 1 was the one initiating the conversation, which may have given away strong clues regarding what player 1 wanted and led the conversation to be over the fruits which player 1 was most interested in. This explanation is supported by that player 1, without any prior knowledge or belief what the opponent wanted, was asked to initialize the conversation given their own fruit endowment and valuations. Due to this, player 1 might have initiated the conversation asking for fruits they valued highest.

As the guesses were used to update each player's belief about what their opponent wants, more correct guesses led to beliefs that were more closely aligned with the opponent's true valuations. Given that the guesses following messages

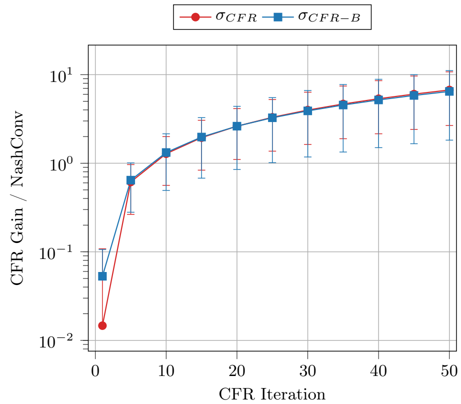

Fig. 8. Average CFR Gain / NashConv over CFR iterations 1 to 50.

1 and 3 were more correct (90% and 80% correct guesses), than those following message 2 (50% correct guesses), it can be concluded that player 2's beliefs were more accurately updated than those of player 1. This difference in belief update success influenced the course of negotiations, as discussed in the following sections.

5.2 Number of Agreements Reached

As shown in Fig. 6, the percentage of agreements reached increased as a function of the message number both with and without, revision of beliefs. That is, the longer the negotiation continued, the more agreements were reached. As expected, no agreements were reached in message 1; as at least two messages, in theory, are needed for an agreement to be made between two players. In message 2, fewer than 1% of messages resulted in an agreement, when prompting the LLM with tone or tone and beliefs. For message 2, no difference in percentage of agreements reached was observed between the two implementations.

For messages 3 and 4, prompting the LLM with tone and beliefs led to a small increase in percentage of agreements reached, compared to prompting it solely with tone. The difference was also more prominent for message 4, than for message 3. These results suggest that incorporating revision of beliefs contributes to an increased number of agreements reached later in a conversation. Whether this trend continues in longer conversations remains a question for future research.

5.3 Value of Trades

In general, trades were more beneficial than not for players, as they resulted is more positive trade payoffs (Table 2) than negative ones (Table 3). That is, the LLM was able to make more trades that were beneficial than not.

The number of trades that were beneficial for player 2 increased, with the addition of revision of beliefs. For player 2, the percentage of positive trade payoffs increased from 59.81% to 77.44% for message 3 and from 48.81% to 67.02% for message 4 (Table 2). The percentage of negative trade payoffs also decreased from 31.10% to 12.79% for message 3 and 32.30% to 16.92% for message 4 (Table 3).

Notably, the same trend was not shown for player 1. The differences in percentage of positive trade payoffs were small and regarded as insignificant (Table 2). The same was concluded about the percentage of negative trade payoffs for player 1 and message 4 (Table 3). The only significant difference for player 1 was the percentage of negative trade payoffs for message 3, which decreased from 32.18% to 25.59%. This result suggests that player 1 was able to turn down more unfavorable trade proposals with revision of beliefs, than without it.

The number of trades that were beneficial for both players moreover increased, with the addition of revision of beliefs. The percentage of trades which resulted in a positive trade payoff for both players increased from 34.09% to 46.13% for message 3 and from 24.01% to 33.13% for message 4 (Table 2). The corresponding percentage of negative trade payoffs for both players also decreased from 11.84% to 5.84% for message 3 and from 6.72% to 3.28% for message 4 (Table 3).

Given that revision of beliefs was used, the resulting trades were more beneficial for player 2 than for player 1. Both the percentage of positive trade payoffs were higher (77.44% compared to 58.92% for message 3 and 67.02% compared to 54.01% for message 4) and the percentage of negative trade payoffs were lower (12.79% compared to 25.59% for message 3 and 16.92% compared to 28.72% for message 4) for player 2 compared to player 1, as presented in Table 2 and 3 respectively.

The difference in the number of beneficial trades for player 1 and 2 can be explained by the difference in success of belief updates (see Sect. 4.1). As belief updates were more successful for player 2, than for player 1, player 2 was more accurately informed about what their opponent wanted. With more accurate beliefs, player 2 could make more informed decisions; which may have resulted in the higher percentage of beneficial trades for that player.

5.4 Effect on the Resulting CFR Strategy

Whereas the incorporation of revision of beliefs increased the percentage of agreements made and the value of trades made for player 2 and both players, the improvement did not transfer to the equilibrium strategies. As shown in Fig. 7, using CFR to find equilibrium strategies resulted in a higher CFR Gain than NashConv both with (σ_{CFR-B}), and without (σ_{CFR}), revision of beliefs. That is, it was possible to find approximate evolutionarily stable strategies with CFR in both cases, that perform better than the uninformed baseline. ESS was also quickly found for each subgame, as only around 100 CFR iterations were needed for both implementations (see Fig. 7).

The difference between using revision of beliefs and not was, however, not significant for the found equilibrium strategy; as shown by the ratio CFR Gain over NashConv in Fig. 8. That is, when combined with game-theoretic solvers, revision of beliefs did not further enhance the LLM performance in the negotiation. This does not mean that revision of beliefs cannot make a difference, but that the difference was not large enough to make a difference in the resulting CFR strategy. The difference may, however, become more prominent in more complex negotiations, than the one utilized here.

As mentioned in Sect. 4 and illustrated in Fig. 4, the subgame—and not the entire game—was solved for equilibrium strategies. That is, for each game initialization, strategies were explored and equilibrium strategies were found independently of each other. Even though the private information—and thus payoffs—of each player was not common knowledge, this information could implicitly leak when solving the subgame for equilibrium strategies using CFR. The private information was implicitly learned as CFR found equilibrium strategies using these payoffs. For this information to remain private, CFR should not be allowed to iterate too many times over each subgame tree. How many iterations that can be made for the private information to remain private, however, remains a question for future research.

Solving the entire game tree may yield different results than those presented here. Due to time constraints, no such implementation was made. Whether solving the entire game tree yields a different result regarding the addition of revision of beliefs, thus remains a question for future research.

5.5 Limitations

The greatest challenge in this study was the number of LLM calls required to perform negotiation simulations to build the game tree. Although the codebase was modified to reduce the number of LLM calls needed, simulations were still computationally expensive after these modifications. To handle this issue, LLM calls were made externally. While this solved the problem of computational power, using external LLM calls became financially expensive, which again limited what was possible and how much could be explored. For this reason, the study was limited to one negotiation domain with 30 game initializations, and a maximum of four messages in each subgame. For the same reason, only one LLM and one belief implementation was tested.

Moreover, the dialogue generation and evaluation relied entirely on the reliability of LLM outputs. For generation, this meant the LLM could convey tone appropriately; for evaluation, it involved correctly identifying agreements and calculating payoffs. While Gemp et al. conducted such reliability analysis [10], this study did not, due to time constraints——though such validation would have strengthened the findings. The deployed model's strong benchmark performance (e.g., MATH, MMLU) relative to those used by Gemp et al. [1,2,10] was considered sufficient to trust its reliability.

As previously noted, only individual subgames—not the full game tree—were solved and evaluated for equilibrium strategies, each tested on the subgame it was derived from. It was not assessed whether strategies generalized across subgames, which would validate the CFR solver's ability to find domain-level optimal strategies. Gemp et al. faced the same limitation [10]. Evaluating generalization to new game initializations would further support such analysis.

Finally, the experimental setup features a simplified and synthesized negotiation domain. While the fruit trading game enables initial testing of belief revision in strategic interactions, it lacks the complexity and uncertainty in real-world adversarial scenarios. This limitation may affect the generalization to more complex negotiation domains. Evaluation of the framework in more realistic domains should therefore be made.

6 Conclusions

To conclude, incorporating revision of beliefs increased the number of agreements reached and the value of trades, given that beliefs were accurately updated. When solving the resulting negotiation game for equilibrium strategies using Counterfactual Regret Minimization (CFR), evolutionarily stable strategies were found both with and without revision of beliefs. However, revision of beliefs did not further improve LLM performance when combined with CFR. Results suggest that belief revision can enhance the strategic capabilities of LLMs, but also indicate that CFR alone is sufficient to produce evolutionarily stable strategies.

Future work should focus on finding and evaluating solutions for different types of games; such as games with more than two players, longer and more complex negotiations, and games where asymmetries in available actions and preferences are found. Different ways to solve games and whether solutions generalize across domains should also be examined. A condition for game theory to become a useful tool, however, requires that LLMs become more efficient to use—to be able to build game trees. Since building game trees is computationally expensive, Bayesian belief revision could also offer a promising solution on its own. Future work should focus on how to perform revision of beliefs and what to maintain beliefs over.

References

1. Llama-3.3-70B-Versatile, `https://console.groq.com/docs/model/llama-3.3-70b-versatile`
2. Anil, R., et al.: PaLM 2 Technical Report (Sep 2023), arXiv:2305.10403
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York, NY, USA, 1 edn. (Aug 2006)
4. Brynielsson, J., et al.: Comparison of Strategies for Honeypot Deployment. In: Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 611–618. ACM, New York, NY, USA (Nov 2023). `https://doi.org/10.1145/3625007.3631602`
5. Chawla, K., et al.: Be Selfish, But Wisely: Investigating the Impact of Agent Personality in Mixed-Motive Human-Agent Interactions. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp.

13078–13092. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.emnlp-main.808
6. Fernquist, J., Kaati, L., Schroeder, R.: Political Bots and the Swedish General Election. In: 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 124–129 (Nov 2018). https://doi.org/10.1109/ISI.2018.8587347
7. Fu, Y., et al.: Improving Language Model Negotiation with Self-Play and In-Context Learning from AI Feedback (May 2023), arXiv:2305.10142
8. Fudenberg, D., Tirole, J.: Game Theory. The MIT Press, Cambridge, MA, USA (Aug 1991)
9. Gelman, A., et al.: Bayesian Data Analysis. Taylor & Francis Inc., New York, NY, USA, 3 edn. (Nov 2013)
10. Gemp, I., et al.: Steering Language Models with Game-Theoretic Solvers (Dec 2024), arXiv:2402.01704
11. Guan, Z., et al.: Richelieu: Self-Evolving LLM-Based Agents for AI Diplomacy. In: 38th Conference on Neural Information Processing Systems. pp. 123471–123497. Curran Associates, Inc., Red Hook, NY, USA (Oct 2024)
12. Harsanyi, J.C.: Games with Incomplete Information Played by "Bayesian" Players, I–III Part I. The Basic Model. Management Science **14**(3), 159–182 (Nov 1967). https://doi.org/10.1287/mnsc.14.3.159
13. Harsanyi, J.C.: Games with Incomplete Information Played by "Bayesian" Players Part II. Bayesian Equilibrium Points. Management Science **14**(5), 320–334 (Jan 1968). https://doi.org/10.1287/mnsc.14.5.320
14. Harsanyi, J.C.: Games with Incomplete Information Played by "Bayesian" Players, Part III. The Basic Probability Distribution of the Game. Management Science **14**(7), 486–502 (Mar 1968). https://doi.org/10.1287/mnsc.14.7.486
15. Herr, N., et al.: Are Large Language Models Strategic Decision Makers? A Study of Performance and Bias in Two-Player Non-Zero-Sum Games (Oct 2024), arXiv:2407.04467
16. Howard, P.N., Woolley, S., Calo, R.: Algorithms, bots, and political communication in the US 2016 election: The challenge of automated political communication for election law and administration. Journal of Information Technology & Politics **15**(2), 81–93 (2018). https://doi.org/10.1080/19331681.2018.1448735
17. Huang, Y.J., Hadfi, R.: How Personality Traits Influence Negotiation Outcomes? A Simulation based on Large Language Models (Nov 2024), arXiv:2407.11549
18. Lanctot, M., et al.: OpenSpiel: A Framework for Reinforcement Learning in Games (Sep 2020), arXiv:1908.09453
19. Liu, X., et al.: AgentBench: Evaluating LLMs as Agents (Oct 2023), arXiv:2308.03688
20. Meta Fundamental AI Research Diplomacy Team (FAIR), et al.: Human-level play in the game of *Diplomacy* by combining language models with strategic reasoning. Science **378**(6624), 1067–1074 (Dec 2022). https://doi.org/10.1126/science.ade9097
21. Nash, J.: Non-Cooperative Games. Annals of Mathematics **54**(2) (Sep 1951). https://doi.org/10.2307/1969529
22. Nash, J.F.: The Bargaining Problem. Econometrica **18**(2), 155–162 (1950)
23. Nisan, N., et al.: Algorithmic Game Theory. Cambridge University Press, New York, NY, USA (Sep 2007)
24. Noh, S., Chang, H.C.H.: LLMs with Personalities in Multi-issue Negotiation Games (May 2024), arXiv:2405.05248
25. Peters, H.: Game Theory: A Multi-Leveled Approach. Springer, Berlin, Germany, 2 edn. (Jun 2015)

26. Peterson, M.: An Introduction to Decision Theory. Cambridge University Press, Cambridge, United Kingdom, 2 edn. (Mar 2017)
27. Rajani, N.F., et al.: Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 4932–4942. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1487
28. Rubinstein, A.: Perfect Equilibrium in a Bargaining Model. Econometrica **50**(1), 97–109 (1982)
29. Smith, J.M., Price, G.R.: The Logic of Animal Conflict. Nature **246**, 15–18 (Nov 1973). https://doi.org/10.1038/246015a0
30. Smith, J.M.: Game Theory and the Evolution of Behaviour. Proceedings of the Royal Society of London. Series B. Biological Sciences **205**(1161), 475–488 (Sep 1979). https://doi.org/10.1098/rspb.1979.0080
31. Tadelis, S.: Game Theory: An Introduction. Princeton University Press, Princeton, NJ, USA (Feb 2013)
32. Wei, J., et al.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. pp. 24824–24837. Curran Associates, Inc., Red Hook, NY, USA (Nov 2022)
33. Xi, Z., et al.: The rise and potential of large language model based agents: a survey. Science China Information Sciences **68**(2), 121101:1–121101:44 (Jan 2025). https://doi.org/10.1007/s11432-024-4222-0
34. Zelikman, E., et al.: STaR: Self-Taught Reasoner Bootstrapping Reasoning with Reasoning. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. pp. 15476–15488. Curran Associates, Inc., Red Hook, NY, USA (Nov 2022)
35. Zhang, Y., et al.: LLM as a Mastermind: A Survey of Strategic Reasoning with Large Language Models. In: First Conference on Language Modeling (Aug 2024), https://openreview.net/pdf?id=iMqJsQ4evS
36. Zinkevich, M., et al.: Regret Minimization in Games with Incomplete Information. In: Proceedings of the 21st International Conference on Neural Information Processing Systems. pp. 1729–1736. Curran Associates, Inc., Red Hook, NY, USA (Dec 2007)