Strategic Steering of Large Language Models via Game-theoretic Action Space Optimization

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

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

Abstract. This paper investigates how large language models can be shaped to act more strategically in text-based negotiation settings. Two action space designs are compared, namely emotional tone-based and explicit offer-based, within a negotiation environment, and outcomes are compared in simulated dialogues. The results show that both approaches improve strategic outcomes compared to a baseline, with tone-based actions yielding higher agreement rates and offer-based actions providing more stable tradeoffs. These findings demonstrate how action space design influences agent behavior, providing insights for deployment of large language models in strategic negotiation scenarios to gain an advantage in, for example, online influence operations.

Keywords: Game theory · Large language models · Action space optimization · Adversarial dialogue · Influence operations

1 Introduction

Complex language is a defining characteristic that sets humans apart from all other species [25]. No other species comes close to matching the structural richness and intricacy of human communication. What distinguishes our communication system most is the vast number of symbols, words, and the sophisticated structure of sentences [13]. Due to this complexity, human language remains one of the most challenging faculties to replicate in machines. The study of computational approaches that enable machines to analyze and process text in ways that emulate human language understanding and use is called natural language processing (NLP) [20]. The first research within NLP in the 1950s, centered on rule-based machine translation and experiments to automate translation from Russian to English [15,32]. Throughout the 1960s and 1970s, NLP systems relied on handcrafted linguistic rules for tasks such as parsing and information extraction [34]. With the rise of large text corpora and greater computing power in the 1980s and early 1990s, statistical language models appeared in the form of n-gram models that estimate the probability of word sequences [6,31]. Through extended research and parallel developments in other fields, the LMs from the 1990s have turned into the large language models (LLMs) we use today.

Only during the last few years, there has been an immense increase of interest and use of LLMs. This increase can be seen in research, where the number of papers released which include “Large Language Model” as keyword has increased from 114 in the year 2020 to 20 900 in 2023 [22]. However, it is not only among researchers this interest has been growing rapidly. Since the release of the chatbot ChatGPT by the AI research company OpenAI in late 2022, ChatGPT had already reached 100 million weekly active users by November 2023, and over 200 million by August 2024 [1].

Despite the rapid development of LLMs over the last years, they still lack in reasoning capabilities compared to humans [19,7]. Many studies have tried, and succeeded, to take LLMs closer to humans with regards to their reasoning capabilities. One such study was made by Gemp et al. in 2024 where they investigated LLMs using a negotiation game [9]. In order to improve the negotiation capabilities of the LLMs, they introduced a framework that allowed for the combination of game-theoretic analysis with natural language. The outcome of the study was that the LLMs that were guided by game-theoretic solvers improved its negotiation capabilities, compared to the LLMs which did not make use of any game theory.

This paper aims to examine the strategic LLM further, by using different actions when steering the LLM. The chosen action spaces are using tones or offers and is compared with each other. This comparison is made in a negotiation game, similar to the one used by Gemp et al., where each model is evaluated against a baseline without any game-theoretic steering. This aims at expanding our knowledge of how LLMs can further enhance their reasoning capabilities. Hence, the research question in this paper is the following: *How does the choice of action space affect large language models’ performance when steered with game-theoretic solvers?*

2 Game-Theoretic Background

This section outlines the game-theoretic foundations for our approach. We cover core concepts from game theory, counterfactual regret minimization (CFR), and the evaluation metrics NashConv and CFR Gain.

2.1 Basics of Game Theory

Game theory is the study of strategic interactions through mathematical models [29]. These interactions are what we will refer to as *games*. All games have some properties which are common. This includes a given starting point from which the game begins [23]. From the starting point, a sequence of moves follow for each player where they have to make one or more decisions. Finally, the player(s) are given a *payoff* depending on the outcome of the game once it has reached a terminal state. Other than this, games can vary widely; they can be cooperative or competitive, involve different numbers of participants, and have perfect or imperfect information.

The concept of a *Nash equilibrium* was introduced by John Nash in his landmark 1950 paper [16], and serves as a solution concept for a wide variety of games. He proved that every finite game (i.e., a game with a finite number of players and actions) has at least one Nash equilibrium in mixed strategies. Before discussing Nash equilibrium any further, we define *best response.* In a two-player game, a strategy σ_1^* is a best response to a strategy σ_2 if the expected payoff when playing σ_1^* is greater or equal to the expected payoff of all other strategies σ_1. If U is a matrix containing the payoffs in normal-form games, the above definition can be written as

$$u_1(\sigma_1^*, \sigma_2) \geq u_1(\sigma_1, \sigma_2) \ \ \forall\, \sigma_1 \in S_1 \text{ such that } \sigma_1 \text{ is a probability vector.}$$

Definition 1. Two strategies (σ_1^*, σ_2^*) is a *Nash equilibrium* if σ_1^* is a best response to σ_2^* and σ_2^* is a best response to σ_1^*.

The definition means that no player can unilaterally gain by deviating from their equilibrium strategy, given that the other player's strategy remains fixed. A *pure Nash equilibrium* is a Nash equilibrium where both σ_1^* and σ_2^* are pure strategies.

In games with *incomplete information*, at least one player does not have all information about the full mathematical structure of the game [24,11,30]. This could, among other things, include the payoffs, strategies or information available to one or more players.

Even small, simple incomplete-information games often lack analytical solutions and require computer algorithms.

2.2 Algorithmic Game Theory

Over the past 25 years, the integration of theoretic computer science into game theory has rapidly given rise to the field of algorithmic game theory (AGT) [26]. AGT differs from classic game theory in several respects. It often examines optimization problems, and its solutions typically consist of optimal approximation bounds rather than the exact solutions sought in classic game theory [27].

Let H be a finite set of *histories*, that is, all possible sequence of actions that may occur in the game. Further, let $A(h) = \{a : (h, a) \in H\}$ be the actions available after a *nonterminal history* $h \in H$. The probability of history h occurring if players choose actions according to the *mixed* strategy σ is denoted $\pi^\sigma(h)$. This can be decomposed into each players contribution to the probability according to

$$\pi^\sigma(h) = \prod_{i \in N} \pi_i^\sigma(h),$$

where N is the set of players. Here, $\pi_i^\sigma(h)$ is the probability that player i, when following strategy σ_i, selects the corresponding actions in h at each of their decision points. In a similar way, $\pi_{-i}^\sigma(h)$ is the product of all other players' contribution to the probability except player i. For a given information set, $I \subseteq H$,

$$\pi^\sigma(I) = \sum_{h \in I} \pi^\sigma(h)$$

is the probability of reaching this information set given σ. Finally, the expected payoff for player i, given a strategy σ, is

$$u_i(\sigma) = \sum_{h \in T} P_i(h) \pi^\sigma(h).$$

This means that the expected payoff is the sum of player i's payoffs, P_i, at every terminal node, T, multiplied by the probability of reaching that node.

2.3 Counterfactual Regret Minimization

The most successful solvers for games with imperfect information are a family of algorithms called counterfactual regret minimization (CFR) [4]. The algorithm was first introduced by Zinkevich et al. in 2007 [35] and has since been modified in several forms to improve performance [4,18,3,5]. In this section, we focus on Zinkevich et al.'s original algorithm [35].

The CFR algorithm is based on action's regret and we begin by describing this concept. Consider an extensive game played repeatedly where σ_i^t is the strategy used by player i in round t. The *average cumulative regret* of player i after τ iterations is

$$R_i^\tau = \frac{1}{\tau} \max_{\sigma_i' \in S_i} \sum_{t=1}^{\tau} \left(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma^t) \right).$$

This equation calculates the difference between the actual payoffs received and the best possible payoffs player i could have received by always playing the best strategy in hindsight, averaged over τ iterations. Let us also define $\bar{\sigma}_i^t$ as the *average strategy* for player i over iterations 1 to τ. The average strategy is the strategy which selects actions with probabilities proportional to the cumulative frequencies with which they were chosen during previous iterations. For each information set I and for each action $a \in A(I)$, the average strategy of that information set is given by

$$\bar{\sigma}_i^t(I, a) = \frac{\sum_{t=1}^{\tau} \left(\pi_i^{\sigma^t}(I) \sigma^t(I, a) \right)}{\sum_{t=1}^{\tau} \pi_i^{\sigma^t}(I)}.$$

This measures how often player i chose action a at information set I in the past, weighted by the probability of reaching I in each iteration.

Next, we define counterfactual regret. The fundamental idea of CFR is to decompose overall regret into additive regret terms, which are defined at each information set in extensive-form games. Zinkevich et al. show that these regrets can be minimized independently at each information set and that overall regret is bounded by the sum of counterfactual regret [35].

Before defining counterfactual regret, let us introduce *counterfactual utility.* The counterfactual utility is the expected payoff given that information set I is reached and every player plays according to strategy σ except player i. It is defined as

$$u_i(\sigma, I) = \frac{\sum_{h \in I, h' \in T} \pi^{\sigma}_{-i}(h) \pi^{\sigma}(h, h') u_i(h')}{\pi^{\sigma}_{-i}(I)}.$$

The term $\pi^{\sigma}(h, h')$ is the probability of moving from history h to h' when playing strategy σ.

The final term to define is the *immediate counterfactual regret.* Let $a \in A(I)$, $\sigma|_{I \to a}$ be the strategy profile identical to σ except that player i plays action a in information set I. The immediate counterfactual regret is then defined as

$$R^{\tau}_{i,\mathrm{imm}}(I) = \frac{1}{\tau} \max_{a \in A(I)} \sum_{t=1}^{\tau} \pi^{\sigma^t}_{-i}(I) \left(u_i(\sigma^t|_{I \to a}, I) - u_i(\sigma^t, I) \right).$$

The above term is player i's regret of a decision at information set I with regards to the counterfactual utility, weighted by the probability that I is reached that round if the player tried to do so. As negative regret is not of interest, $R^{\tau,+}_{i,\mathrm{imm}}(I)$ is defined as the positive immediate counterfactual regret by $\max(0, R^{\tau}_{i,\mathrm{imm}}(I))$. This leads to the first theorem which gives an upper bound on the overall regret.

Theorem 1. The overall regret is upper bounded by the sum of positive immediate counterfactual regret, that is

$$R^{\tau}_i \leq \sum_I R^{\tau,+}_{i,\mathrm{imm}}(I).$$

Zinkevich et al. prove the above theorem in their paper [35]. The theorem states that minimizing the immediate counterfactual regret also minimizes the overall regret. Hence, an approximate Nash equilibrium can be found by minimizing the immediate counterfactual regret. The next question is, how to minimize the immediate counterfactual regret. As mentioned above, it can be minimized at each information set independently. Especially, for all I and for all $a \in A(I)$, we define

$$R^{\tau}_i(I, a) = \frac{1}{\tau} \sum_{t=1}^{\tau} \pi^{\sigma^t}_{-i}(I) \left(u_i(\sigma^t|_{I \to a}, I) - u_i(\sigma^t, I) \right)$$

and $R^{\tau,+}_i(I, a) = \max(0, R^{\tau}_i(I, a))$. A strategy for player i at iteration, $\tau + 1$ is

$$\sigma^{\tau+1}_i(I, a) = \begin{cases} \frac{R^{\tau,+}_i(I,a)}{\sum_{a \in A(I)} R^{\tau,+}_i(I,a)} & \text{if } \sum_{a \in A(I)} R^{\tau,+}_i(I, a) > 0, \\ \frac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

According to the above strategy, each action a is selected with probability proportional to its positive counterfactual regret for not playing a. That is, the more a player would counterfactually regret not playing an action, the more likely that action is to be played. If no action has positive counterfactual regret, the player chooses uniformly at random among the available actions.

Zinkevich et al. concludes that if players choose actions according to the above equation, it can be used in self-play to compute a Nash equilibrium. Self-play is a way of learning optimal policies by playing against itself without supervision [2]. As the immediate counterfactual regret is an upper bound for the average overall regret, if the immediate counterfactual regret at iteration τ is less than ϵ in a zero-sum game, then $\bar{\sigma}^{\tau}$ is a 2ϵ Nash equilibrium [35].

2.4 Evaluation Metrics

To evaluate an approximate Nash equilibrium, we define two key metrics. Similar measures were used by Gemp et al. to evaluate if the LLMs became more strategic under the influence of algorithmic solvers [9]. The first measure is called *CFR Gain.* To calculate CFR Gain, a scenario where both players are playing an uninformed baseline is used. That is, neither player is utilizing the optimized strategy found by the CFR solver. From this scenario, one asks how much either player would gain if they switched from playing the uninformed baseline to the CFR strategy. This is measured by the increase or decrease in payoff. A positive value of CFR Gain, indicates that the player is gaining by switching to the CFR strategy compared to an uninformed baseline.

The other measure is *NashConv.* This is computed through a scenario where both players by default are playing the CFR strategy. One then asks, how much either player would gain by switching to any alternative pure or mixed strategy, compared to the CFR strategy. In other words, NashConv is a measure of how far the CFR strategy is from an equilibrium [14]. A high CFR Gain, relative to NashConv, means that the mixed strategy calculated by the CFR solver approximately satisfies the condition for an evolutionarily stable strategy [28].

3 Methods: Optimizing Large Language Model Negotiation

This section describes our methods for optimizing a large-language-model negotiator using counterfactual regret minimization. We specify the task and environment and construct two action spaces: (i) tones and (ii) explicit offers.

3.1 The Chat Game

The environment in which the simulations were performed, `OpenSpiel`, is an open codebase developed by Google DeepMind [9]. `OpenSpiel` is an environment for conducting research within reinforcement learning and planning in games [17]. Methods to analyze games and evaluation metrics are also available through `OpenSpiel`. The environment also provides a negotiation game framework, called `chat_game`, which supports several domains. In this paper, `trade_fruit` domain, was used to address the research question. An overview of the game will be presented before describing the two experimental set-ups.

In the fruit trading domain, a game is played between two players negotiating fruit trades. Each player is initially given a set of fruits and valuations of them. In this study, the fruits used were apples, bananas, and oranges. For each player, the initial amount of each fruit was randomly chosen from a discrete uniform distribution between 1 and 2, $U\{1,2\}$. Each player was also given a valuation of each fruit, randomly assigned from $U\{0,4\}$. The endowment and valuations of fruits are private information to the players, they do not know how many of each fruit their opponent has or how she values them. However, the distribution from which the values are drawn are public information and known by both players. An example of the private information can be seen in Fig. 1.

Example private information

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

Fruit Valuations:
apples: 1
bananas: 2
oranges: 4
```

Fig. 1. Example of private information for a player in the trading fruit game.

After the initialization described above, player 1 sends her first message. This is done by an LLM prompted with instructions on how to reply, example messages, as well as the private information of player 1. Player 2 is then asked to respond to the first message of player 1 through an LLM prompted with instructions on how to reply, example conversations, the private information of player 2, and the first message from player 1. The LLM is asked to either accept the trade offer or propose a counter-trade in its response. With the reply of player 2, the LLM is also prompted with a specific action to use, which is to be optimized later. The different actions are described in Sect. 3.2 and 3.3. The game ends either when an agreement has been made or when a predefined maximum number of messages has been sent. Until this condition has been met, the players take turns responding to each others messages. An example conversation can be seen in Fig. 2.

After each message, the conversation history is sent to an LLM asked to summarize the discussion up until that point. An LLM is then prompted with this summary in order to determine if the two players have reached an agreement. If the players are assumed to have reached an agreement, or if the maximum

Example conversation

```
############################
Trade Proposal Message:
from: Leif
to: Ann
############################

Hi Ann,

I propose trading you 1 banana for 1 orange. I think this is a fair exchange.

Let me know if you agree.

Best,

Leif

############################
Trade Proposal Message:
from: Ann
to: Leif
############################

Hi Leif,

Thanks for reaching out. I only have 1 orange and I do not want to give it up. Would
you be willing to take an apple instead? I would like to trade you 1 apple for 1
banana.

Does that work?

Best,

Ann
```

Fig. 2. Example conversation between players Leif and Ann with a counter-proposal. In this example one could imagine Ann having the private information as shown in 1.

number of steps has been reached, the conversation together with the private information of both players, is sent to an LLM asked to calculate the payoffs of the players. If the conversation has reached the maximum number of steps but no agreement has been reached, the rewards are calculated based on the counter-proposal last sent. If this offer is considered advantageous to the receiving player, the payoffs are calculated based on this proposal. The payoffs are defined as the value of the player's fruit endowment times the fruit valuation after the trade minus before the trade. If the last message is not considered advantageous to the receiver, both players receive a payoff of 0.

An overview of the LLM calls made in the fruit trading game can be seen in Fig. 4. All LLM calls used the same model, the only difference between LLM calls are the prompts. The LLM used in all simulations was from the Llama 3 model family. More specifically the model was *Llama-3.3-70B-Versatile* which is an auto-regressive language model with 70 billion parameters trained using supervised fine-tuning and reinforcement learning with human feedback [21]. This was the largest model deemed practical to use in order to ensure the simulations remained manageable in terms of time and cost.

3.2 Tones as Actions

The first implementation is similar to that of Gemp et al. [9]. In order to steer LLMs through game-theoretic solvers, they implemented actions in the form of tones. During the message generation step from Fig. 4 the LLM is asked to generate the answer using a specific tone. The four tones used were *calm*, *assertive*, *submissive*, and *any tone*. The last option, *any tone*, was used to evaluate the solution and the LLM was not prompted with any request about a specific tone.

Calculating payoff for player Ann

```
Value before the trade:
2 * 1 + 2 * 2 + 1 * 4 = 10

Ann receives 1 banana and gives away 1 apple.

Value after the trade:
1 * 1 + 3 * 2 + 1 * 4 = 11

Final payoff for Ann: 11 - 10 = 1
```

Fig. 3. Payoff calculation for player Ann based on the private information in 1 and the trade proposal from the conversation in 2.

In order to find an improved strategy through the choice of tone, at each stage of the game apart from the first message, a separate response was generated for each tone.

As LLM generations are stochastic and will produce different texts every time, a parameter called `num_llm_seed` controlled the number of responses generated per tone. This was to avoid a random text generation to influence the result too much. In this paper, similar to Gemp et al., two responses were generated for each tone [9]. All these answers form a game tree in extensive form. An example of a game tree is illustrated in Fig. 5 where there is one initial message, then four tones each having a chance node where two random messages are created per tone. As can be seen in the game tree, the action space for the player is the set of available tones. A strategy is therefore a probability distribution over the tones, which is to be optimized through an algorithmic game theory solver. If an agreement is reached on one of the branches in the game tree, that branch will terminate at the step of the agreement and no further messages will be generated for that branch. When all messages have been generated, the payoffs are calculated for all terminal nodes as described in Sect. 3.1. In this study the maximum number of messages was set to two, that is one per player. The reason for this was the larger game trees generated in the action space with offers, which required more computational power for LLM calls. Therefore, longer conversations were not feasible to implement. When the game tree had been generated and all payoffs calculated, the CFR solver from `OpenSpiel` [17] was applied on the game as described in Sect. 2.3. CFR iterated over the sub-game tree starting after player 1's initial message 100 times and the updated mixed strategy over the available actions for player 2, along with other evaluation metrics, were reported for every fifth iteration. For more information about the evaluations, see Sect. 3.4. In total, 32 separate games were played.

3.3 Offers as Actions

In the second implementation, the action space was set to contain possible trade combinations, henceforth referred to as offers. The combinations of trades in

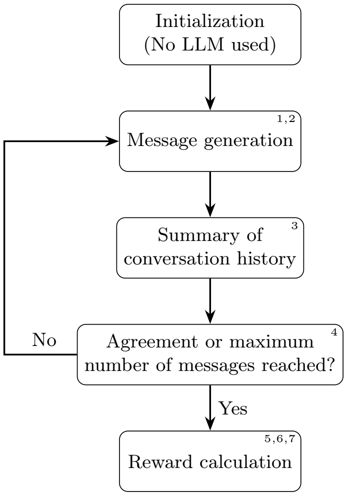

Fig. 4. Overview of the LLM calls and the structure of the fruit trading game. The numbers in the upper right corner of each box represent which LLM prompt is being used at what stage.

the action space were all trades where one type of fruit was exchanged against another type of fruit. With the condition that either one or two of each fruit type was proposed to give or receive. As the maximum endowment of a single fruit was set to two, one could not ask for three or more pieces of a single fruit. Offers where one player either gave or received no fruit were also omitted, as they would never be advantageous for both players. In order to limit the size of the game tree, trades where a combination of fruit types were offered or asked for was not included in the action space. All valid trades where one player receives one apple can be seen in Fig. 6, together with examples of invalid trade suggestions. Apart from the possible fruits offers, the action space also included an option which is for the player to propose *any offer*. This means that the total size of the action space was 25. As with the *any tone* option in the baseline implementation, the LLM was not given any offer at all when playing the *any offer* action. As the action space is changed, and enlarged, the game tree in the updated version will look different from that in Fig. 5. The updated game tree can be seen in Fig. 7. Every action still has two LLM seeds as chance nodes and the maximum number of messages was set to two, before calculating the payoffs. Each game was iterated 100 times with the CFR solver and a total of 32 games were simulated.

3.4 Evaluation

The two implementations were evaluated separately. Firstly, two metrics used by Gemp et al. [9] were used to determine whether the improved strategy which

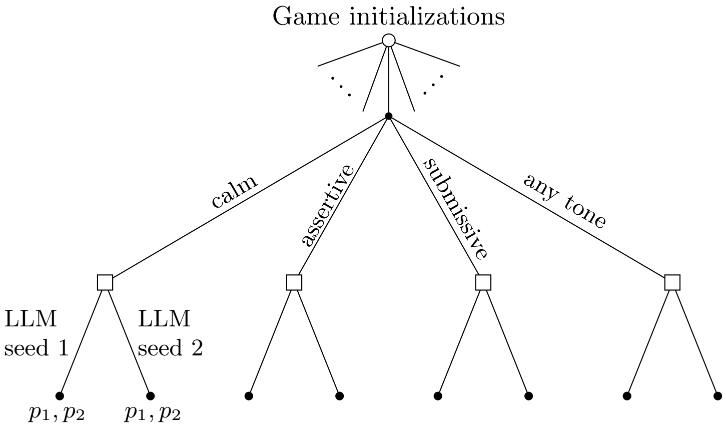

Fig. 5. An example of a game tree with tones as actions. The game initialization (○) contains a random setup of fruit endowment and fruit valuation before the initial message is sent by the first player. The initial message does not include any specific action. Then the four possible tones for player 2 are all followed by a chance node (□) with two LLM seeds. One message is generated for each seed. Following the final message the payoffs are calculated for both players, p_1 and p_2.

Example trades

Valid trades	Invalid trades
Receive 1 apple and give 1 banana	Receive 1 apple and give 0 bananas
Receive 1 apple and give 2 bananas	Receive 1 apple and give 1 apple
Receive 1 apple and give 1 orange	Receive 1 apple and give 2 apples
Receive 1 apple and give 2 oranges	Receive 1 apple and give 3 oranges

Fig. 6. All valid trades where one player receives one apple are shown to the left. In the right column, examples of invalid trades are listed. Trades are invalid if they receive or give anything else then 1 or 2 of a fruit, or if it asks for the same fruit as it wants to gives up.

utilized the optimal policy from the CFR solver performed better than an uninformed baseline. The uninformed baseline always played the *any tone* or *any offer* action, depending on the action space version. This baseline was compared to a model following the equilibrium strategy computed by the CFR solver. The two measures, CFR Gain and NashConv, are described in Sect. 2.4.

To evaluate in which of the two implementations most agreements were reached, the percentage of agreements reached was calculated. To further deepen the analysis, it was examined at what stage a potential agreements were reached, after two or three messages. This was initially done by calculating the percentage of conversations where the LLM identified that the players had reached an agreement after two messages. This refers to the LLM in the penultimate step in Fig. 4, which is asked to identify if an agreement has been made. For the LLM

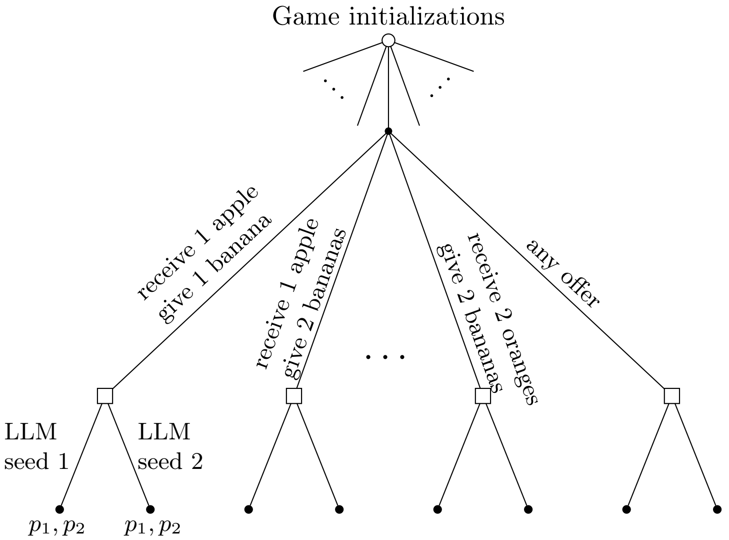

Fig. 7. An example of a game tree with offers as actions. The total number of actions are 25 but only four are shown in the game tree.

to register an agreement, the player receiving the first message must accept it directly. Any other response should not trigger the LLM to report an agreement. In the second approach, the number of agreements was derived from counting the number of conversations that ended in a non-zero payoff for at least one player. This corresponds to an agreement being made following a hypothetical third message. Recall from Sect. 3.1 that the rewards are set to zero for both players if the last trade offer is not advantageous for the receiving player. Otherwise, it is calculated based on the final offer.

Finally, it was of interest to measure the outcome of the trades. This is referring to how often the players gained or lost value through the negotiated trade. These metrics give an indication on which of the implementations where the players managed to come to more favorable agreements. This was evaluated through four measures. The first two measures were the percentage of agreements where at least one player received a negative payoff, and where both players received a negative payoff was calculated. A negative payoff corresponds to the player losing out on the trade and having a lower valuation of her fruits after the trade compared to before. When calculating the number of negative rewards for at least one player, the conversations which resulted in negative rewards for both players were also included. Finally, the percentage of conversations ending with at least one positive payoff or two positive payoffs was also calculated to examine how many conversations ended with a positive outcome for either player or both players respectively.

4 Results

This section provides the results from the experiments described in Sect. 3. First, the performance of the game-theoretic solver is presented. Next, the number of agreements made in the game are covered. Finally, the value of the trades are presented.

4.1 Effect of Game-Theoretic Solvers

The effect of game-theoretic solvers was evaluated based on the two measures, NashConv and CFR Gain, as described in Sect. 2.4. In Table 1 one can see the reported values after 20 CFR iterations (compared to 10 iterations reported by Gemp at al. [9]). The average values for NashConv and CFR Gain through the CFR iterations are shown in Fig. 8.

Table 1. Average NashConv and CFR Gain with corresponding standard deviation after 20 iterations of CFR solver for both implementations.

Actions	Average NashConv	Average CFR Gain
Tones	0.0031 ± 0.0019	0.0193 ± 0.0169
Offers	0.0109 ± 0.0062	0.0321 ± 0.0189

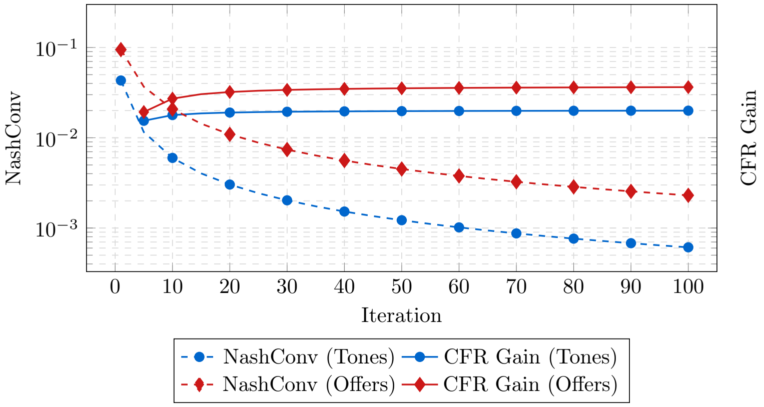

Fig. 8. Log-scale plot of the average NashConv and CFR Gain over the first 100 iterations.

The ratio of CFR Gain divided by NashConv for the first 100 iterations is presented in Fig. 9. It was calculated as the average CFR Gain value at every iteration divided by the average NashConv value. In Table 2 the exact values for

Table 2. Ratio of CFR Gain divided by NashConv after 20 iterations.

Actions	Ratio
Tones	6.28 ± 6.81
Offers	2.95 ± 2.40

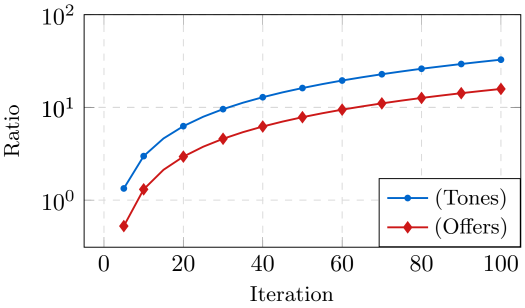

Fig. 9. Ratio of CFR Gain divided by NashConv over the first 100 iterations for the two scenarios.

the ratios after 20 iterations of CFR solver are reported. If this value is greater than one, CFR Gain is larger than NashConv and the optimized CFR strategy approximately satisfies the conditions for an evolutionary stable strategy, see Table 2.4.

4.2 Agreement Frequency

The number of agreements between the players was also measured. Recall from Sect. 3.4 that this was done with two different measures. The first one was the percentage of conversation that the LLM reported had reached an agreement after two messages, that is the receiver accepting the trade offer from the first player. The percentages are shown in Table 3.

Table 3. Percentage of agreements reported by LLM after two messages.

Actions	Agreements after 2 messages
Tones	2.93%
Offers	0.62%

The second measure was the number of conversations resulting in a non-zero reward for at least one player. This corresponds to the first player receiving a countertrade which she would deem beneficial and agree to trade after a hypo-

Table 4. Percentage of conversations ending with a payoff of zero for both players. All other conversations are assumed to have reached an agreement following a hypothetical third message.

Actions	Zero Payoff	Non-Zero Payoff
Tones	25.46%	74.54%
Offers	36.16%	63.84%

thetical third message. In Table 4, the percentage of trades resulting in a zero payoff and non-zero payoff after three possible conversation steps are reported.

4.3 Trade Value Analysis

The final measures which were examined was the value of the trades. The percentage of conversations ending in at least one negative payoff as well as those resulting in two negative payoffs are presented in Table 5. This corresponds to either one or both players making an exchange which is not beneficial. The percentage of conversations where at least one player received a positive reward, as well as the percentage of conversations where both players ended with a positive payoff, are presented in the last two columns. Remember that a conversation can include both a positive and a negative reward and the total percentage in Table 5 will therefore not add to 100.

Table 5. Percentage of conversations ending with at least one negative payoff, two negative payoffs, at least one positive payoff and two positive payoffs for both settings.

Actions	≥ 1 negative	2 negative	≥ 1 positive	2 positive
Tones	55.40%	10.35%	58.38%	10.27%
Offers	48.50%	5.35%	49.90%	6.47%

5 Discussion

This section provides a discussion of the results presented in Sect. 4. The results are discussed in the same order as they are presented in Sect. 4. At the end of this section the limitations are presented.

5.1 Effect of Game-Theoretic Solvers

Already after a few iterations, both implementations get a CFR Gain that is larger than their NashConv, see Sect. 4.1. This indicates that the strategy found is, approximately, an evolutionarily stable strategy. This means that following the equilibrium strategy, on average, leads to a larger payoff for the player. This is similar to the results Gemp et al. obtained for the tones action space [9].

Comparing the two approaches in Fig. 8, show that the overall appearance and trend are very similar. Both CFR Gains converge within the first 50 iterations, which means that the gain from switching from the uninformed *any* action to the CFR strategy is constant after 50 iterations. How much either player would gain by switching from the CFR strategy to any other available strategy, NashConv, is still decreasing even after 100 iterations. However, the NashConv value is lower than the CFR Gain value already after a few iterations. For both NashConv and CFR Gain, the values for the implementation using offers as actions were higher than that using tones. This means that the gain from switching from the *any offer* action to the CFR strategy is larger, but also the gain switching from the CFR strategy to some other strategy is larger. This difference is most likely due to the increased action space size of offers. As the offers action space is more than six times larger, 25 actions compared to 4 actions, the amount of available strategies is also larger. With a larger set of strategies it is more likely that the optimized strategy will outperform the baseline as it has more actions to choose from and can fine-tune decisions more precisely. It also reduces the probability that the baseline *any* action is included in the CFR strategy which would further increase CFR Gain. Regarding the NashConv value, a similar argument can be made. With more actions to choose from, the gain from switching from the CFR strategy to any other available strategy is likely to be larger.

The ratio between CFR Gain and NashConv from the two scenarios in Fig. 9 show that they once again have similar trends. The value for the tones actions is slightly higher, indicating a larger CFR Gain relative to NashConv compared to the offers actions. As the action space is larger for the offers implementation, it might require more iterations for the larger action space to reach a similar ratio compared to the tones. Recall that CFR is an iterative self-play algorithm and a larger action space could therefore lead to more iterations needed to reach the same results.

5.2 Agreement Frequency

Table 3 shows the percentage of conversations that have reached an agreement after two messages. For both approaches, the percentage is very low. This means that the second player is very unlikely to accept the offer after the first message. Based on real-world conversations, this seems like a reasonable result. Most trades are usually not accepted after the first offer, and as LLMs are trained on online text which probably includes real-world trades the LLM tends to mimic this behavior. Another reason for the low agreement percentage after two messages is that the players could have the same endowment and/or valuation of fruits. In this case, they are both interested in the same fruits and are therefore less likely to come to an agreement. In theory, there should not be a major difference in agreement percentages after two messages for the two implementations. Player 1 is not prompted with any specific action to take, and for an agreement to be registered, player 2 has to respond with a confirmation of the agreement which will make the prompted action for player 2 insignificant. The action space only has an influence in the case of a counterproposal from player 2 and should therefore not influence the agreement percentages after two messages. The difference between the implementations might be a coincidence that the fruit valuation and endowment, or the first message generated, were more favorable for agreements in the tones action space. As 32 games, and thus also 32 first messages, were generated this is not an unthinkable scenario. Another reason for the difference could be that the offers implementation is prompted with a trade to offer (in all messages apart from the action *any offer*). This could make the LLM more likely to propose the trade it has been prompted with, even if it is asked to accept the trade if it finds it favorable.

After a hypothetical third message, which is only an accept/decline decision taken by the reward LLM, a majority of the conversations reached an agreement, see Table 4. This metric cannot be directly compared with the values after two messages from Table 3, as it is evaluated in a different way. Firstly, there is no third message generated for the LLM to evaluate if the conversation has reached an agreement. Instead the LLM that calculates the rewards is asked to only calculate the rewards if the trade proposal is advantageous for the player receiving the offer. If not, a payoff of zero should be given to both players. Based on this approach, a conversation should never end with two negative rewards after the third message as the LLM should then decline the offer and apply a payoff of zero to both players. However, as one can see in Table 5, there are conversations ending with two negative rewards. This indicates that the LLM sometimes does not manage to correctly determine if the proposal is advantageous or not.

From Table 4, one can see that 25% of the conversations ended in zero payoff for both players using the tones actions. The same value for the offers action is 36%. It was expected that the offers actions would lead to more conversations not ending in an agreement. As the LLM is prompted with an exact offer to propose, it does not consider the proposal made by player 1 in the first message. Using tones as actions however, allows the LLM to make a counter-offer based on the first player's proposal. The second offer is therefore more likely to be accepted by the LLM during reward calculations when using tones as actions.

5.3 Trade Value Analysis

The final results to discuss are the outcome of the trades. In Table 5 one can see the percentage of different outcomes for the conversations. The implementation with offers as actions ended in a zero payoff for both players more often than the tones actions, as can be seen in Table 4. Because of this, the values for the offers actions are generally lower than those for tones actions in Table 5. Notably, the percentage of both 2 negative and 2 positive outcomes are relatively large for tones actions compared to offers. As discussed in the previous section, as the tones actions have more freedom regarding what offer to suggest, it is likely that it can propose a more advantageous counter-offer based on the first player's message. This is most likely the reason for the higher percentage of conversations ending with 2 positive rewards.

6 Limitations

The largest limitation is concerning the computational power of the LLM calls. As the necessary resources to run the LLM locally over a reasonable time frame was not available, external LLM calls had to be used. Even if it resulted in quicker LLM responses, it limited the number of games that could be ran due to the price of buying LLM calls. It also limited the length of the conversations for the same reason. Conversations with four messages with all 26 offers action was not feasible, neither from a time nor budget perspective.

Since LLM text generations are stochastic, the same prompt can produce different outputs across multiple runs. As action outcomes depend on these generated texts, their evaluation is influenced by this randomness. This issue can be reduced by increasing the number of LLM seeds per action, which allows averaging over more generated responses. However, due to time and budget constraints, two seeds per action were used in this study. To keep the action space with offers at a reasonable size, the game set-up was limited to a maximum of two fruits per type. This meant that the players often had similar fruit endowments. To create more variety in the negotiations, both more fruits of each type as well as more types of fruits could be added in the future.

7 Conclusion

This paper examined how different action spaces affect large language models' performance when steered with game-theoretic solvers. This was investigated through a negotiation game previously employed by Gemp et al. [9]. First, the game was re-implemented together with the previously used action space of tones. A new action space was then implemented which utilized offer combinations as actions. The two implementations were compared through a variety of metrics, including (*i*) whether the optimized strategy converged to an approximate evolutionarily stable strategy, (*ii*) the percentage of agreements reached, and (*iii*) the final outcome of the trades.

The results showes that both implementations, tones and offers, meet the approximate conditions for an evolutionarily stable strategy. This proves that using either action space, the LLM benefits from following the guidance of the CFR strategy and both actions have proven useful and interpretable by the LLMs. The implementation using tones as actions reaches more agreements compared to using offers as actions. This difference is notable after a hypothetical third message where the tone actions reached almost 10 percentage points more agreements than the offer actions. This is likely because the tones actions are free to respond with any trade-proposal it finds fitting. It could therefore tailor the proposal to better respond to the initial message from the first player. The implementation using offers as actions is given an offer to propose in the prompt.

Therefore, it does not take the first player's offer into account when deciding what counter-proposal to suggest.

From the results in this paper, the LLM seems to calculate the correct payoff for a vast majority of conversations. However, it is not flawless, as both implementations report cases where both players receive a negative reward which should instead have been reported as a non-agreement by the LLM. This indicates that there are ways to improve and further deepen the work of this study, as discussed in Sect. 7.1. The overall results and discussion suggest that both implementations show similar performance and that the choice of action space does not affect the LLMs significantly. To conclude, both using tones and offers as actions work when finding optimal strategies using game-theoretic solvers.

7.1 Future Work

For future research, longer conversations should be investigated. Ideally, the conversations would be able to run until an agreement is reached or until they decide not to make a trade. This would require more computational power, but as LLMs get more advanced and effective, this should be feasible in the future. The set of available fruits and the number of each type of fruit could also be increased in future studies to create better opportunities for favorable trades.

Apart from modifications to this specific game, it would also be of interest to examine different negotiation topics. The negotiation capabilities of LLMs could be compared between domains to investigate in which type of negotiations the LLM performs best. Another interesting direction would be to perform negotiations between LLMs and humans, using a similar setup to this game, and examining it from a game-theoretic perspective. Previous studies have shown that humans do not always act rationally in decision games, but rather choose to punish their opponent [10,12]. This would provide an interesting evaluation of the LLMs.

Future research could also include other methods to improve the reasoning. Using game-theoretic solvers is only one way to enhance the LLMs' capabilities. Using chain-of-thought, or providing feedback to the LLM, are two examples of methods that have shown promising signs [33,8]. These methods could be examined through similar negotiation games and compared to the CFR approach used in this paper.

References

1. Babu, J.: OpenAI says ChatGPT's weekly users have grown to 200 million. Reuters (Aug 2024), https://www.reuters.com/technology/artificial-intelligence/openai-says-chatgpts-weekly-users-have-grown-200-million-2024-08-29/
2. Bai, Y., Jin, C., Yu, T.: Near-optimal reinforcement learning with self-play. In: Advances in Neural Information Processing Systems 33. NeurIPS 2020, Curran Associates Inc., Red Hook, NY, USA (2020). https://doi.org/10.5555/3495724.3495906

3. Bowling, M., Burch, N., Johanson, M., Tammelin, O.: Heads-up limit hold'em poker is solved. Communications of the ACM **60**(11), 81–88 (Oct 2017). https://doi.org/10.1145/3131284
4. Brown, N., Lerer, A., Gross, S., Sandholm, T.: Deep Counterfactual Regret Minimization. In: Proceedings of the 36th International Conference on Machine Learning. pp. 793–802. PMLR (May 2019), https://proceedings.mlr.press/v97/brown19b.html
5. Brown, N., Sandholm, T.: Solving imperfect-information games via discounted regret minimization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1829–1836 (2019)
6. Brown, P.F., Della Pietra, S.A., Della Pietra, V.J., Mercer, R.L.: Class-based n-gram models of natural language. Computational Linguistics **18**(4), 467–479 (1992)
7. Costarelli, A., Allen, M., Hauksson, R., Sodunke, G., Hariharan, S., Cheng, C., Li, W., Clymer, J., Yadav, A.: GameBench: Evaluating Strategic Reasoning Abilities of LLM Agents (2024). https://doi.org/10.48550/ARXIV.2406.06613
8. Fu, Y., Peng, H., Khot, T., Lapata, M.: Improving Language Model Negotiation with Self-Play and In-Context Learning from AI Feedback (2023). https://doi.org/10.48550/ARXIV.2305.10142
9. Gemp, I., Patel, R., Bachrach, Y., Lanctot, M., Dasagi, V., Marris, L., Piliouras, G., Liu, S., Tuyls, K.: Steering Language Models with Game-Theoretic Solvers (Dec 2024). https://doi.org/10.48550/arXiv.2402.01704, arXiv:2402.01704
10. Hagen, E.H., Hammerstein, P.: Game theory and human evolution: A critique of some recent interpretations of experimental games. Theoretical Population Biology **69**(3), 339–348 (May 2006). https://doi.org/10.1016/j.tpb.2005.09.005
11. Harsanyi, J.C.: Games with Incomplete Information. In: Götschl, J. (ed.) Evolution and Progress in Democracies, pp. 43–55. Springer Netherlands, Dordrecht (1994). https://doi.org/10.1007/978-94-017-1504-1_2
12. Hewig, J., Kretschmer, N., Trippe, R.H., Hecht, H., Coles, M.G.H., Holroyd, C.B., Miltner, W.H.R.: Why humans deviate from rational choice. Psychophysiology **48**(4), 507–514 (Apr 2011). https://doi.org/10.1111/j.1469-8986.2010.01081.x
13. Hurford, J.R.: Human uniqueness, learned symbols and recursive thought. European Review **12**(4), 551–565 (Oct 2004). https://doi.org/10.1017/S106279870400047X
14. Jin, X., Wang, Z., Du, Y., Fang, M., Zhang, H., Wang, J.: Learning to Discuss Strategically: A Case Study on One Night Ultimate Werewolf (Jan 2025). https://doi.org/10.48550/arXiv.2405.19946, arXiv:2405.19946
15. Jones, K.S.: Natural language processing: a historical review. Current Issues in Computational Linguistics: in Honour of Don Walker pp. 3–16 (1994)
16. Kreps, D.M.: Nash Equilibrium. In: Eatwell, J., Milgate, M., Newman, P. (eds.) Game Theory, pp. 167–177. Palgrave Macmillan UK, London (1989). https://doi.org/10.1007/978-1-349-20181-5_19
17. Lanctot, M., Lockhart, E., Lespiau, J.B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., Hennes, D., Morrill, D., Muller, P., Ewalds, T., Faulkner, R., Kramár, J., Vylder, B.D., Saeta, B., Bradbury, J., Ding, D., Borgeaud, S., Lai, M., Schrittwieser, J., Anthony, T., Hughes, E., Danihelka, I., Ryan-Davis, J.: OpenSpiel: A Framework for Reinforcement Learning in Games (Sep 2020). https://doi.org/10.48550/arXiv.1908.09453, arXiv:1908.09453

18. Lanctot, M., Waugh, K., Zinkevich, M., Bowling, M.: Monte carlo sampling for regret minimization in extensive games. In: Advances in Neural Information Processing Systems 22. p. 1078–1086. NIPS 2009, Curran Associates Inc., Red Hook, NY, USA (2009). https://doi.org/10.5555/2984093.2984215
19. Lee, S., Sim, W., Shin, D., Seo, W., Park, J., Lee, S., Hwang, S., Kim, S., Kim, S.: Reasoning Abilities of Large Language Models: In-Depth Analysis on the Abstraction and Reasoning Corpus. ACM Transactions on Intelligent Systems and Technology p. 3712701 (Jan 2025). https://doi.org/10.1145/3712701
20. Liddy, E.D.: Natural language processing. In: Dekker, M. (ed.) Encyclopedia of Library and Information Science. Marcel Dekker, Inc., New York, 2 edn. (2001), https://surface.syr.edu/istpub/20
21. Llama Team: Llama 3 (2024), https://www.llama.com/models/llama-3/
22. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., Mian, A.: A Comprehensive Overview of Large Language Models (Oct 2024). https://doi.org/10.48550/arXiv.2307.06435, arXiv:2307.06435
23. Owen, G.: Game Theory. Emerald Group Publishing (2013)
24. Peters, H.: Game Theory: A Multi-Leveled Approach. Springer Texts in Business and Economics, Springer Berlin Heidelberg : Imprint: Springer, Berlin, Heidelberg, 2nd ed. 2015 edn. (2015)
25. Premack, D.: Is Language the Key to Human Intelligence? Science **303**(5656), 318–320 (Jan 2004). https://doi.org/10.1126/science.1093993
26. Roughgarden, T.: An algorithmic game theory primer. In: Proceedings of the 5th IFIP International Conference on Theoretical Computer Science (TCS). An invited survey (2008)
27. Roughgarden, T.: Algorithmic game theory. Communications of the ACM **53**(7), 78–86 (2010)
28. Smith, J.M., Holliday, R.: 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
29. Straffin, P.D.: Game Theory and Strategy, vol. 36. MAA (1993)
30. Tadelis, S.: Game Theory: An Introduction. Princeton University Press, Princeton, New Jersey, USA (2013)
31. Wang, Z., Chu, Z., Doan, T.V., Ni, S., Yang, M., Zhang, W.: History, development, and principles of large language models: an introductory survey. AI and Ethics (Oct 2024). https://doi.org/10.1007/s43681-024-00583-7
32. Weaver, W.: Translation. Technical Report No. 19, Rockefeller Foundation, New York (1949), memorandum by Warren Weaver
33. Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q.V., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 24824–24837. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
34. Winograd, T.: Understanding natural language. Cognitive Psychology **3**(1), 1–191 (1972)
35. Zinkevich, M., Johanson, M., Bowling, M., Piccione, C.: Regret minimization in games with incomplete information. In: Advances in Neural Information Processing Systems 20. p. 1729–1736. NIPS 2007, Curran Associates Inc., Red Hook, NY, USA (2007). https://doi.org/10.5555/2981562.2981779