

Article

Effect of Private Deliberation: Deception of Large Language Models in Game Play

Kristijan Poje ^{*}, Mario Brcic , Mihael Kovac  and Marina Bagic Babac 

Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia; mario.brcic@fer.hr (M.B.); mihael.kovac@fer.hr (M.K.); marina.bagic@fer.hr (M.B.B.)

* Correspondence: kristijan.poje@fer.hr

Abstract: Integrating large language model (LLM) agents within game theory demonstrates their ability to replicate human-like behaviors through strategic decision making. In this paper, we introduce an augmented LLM agent, called the private agent, which engages in private deliberation and employs deception in repeated games. Utilizing the partially observable stochastic game (POSG) framework and incorporating in-context learning (ICL) and chain-of-thought (CoT) prompting, we investigated the private agent's proficiency in both competitive and cooperative scenarios. Our empirical analysis demonstrated that the private agent consistently achieved higher long-term payoffs than its baseline counterpart and performed similarly or better in various game settings. However, we also found inherent deficiencies of LLMs in certain algorithmic capabilities crucial for high-quality decision making in games. These findings highlight the potential for enhancing LLM agents' performance in multi-player games using information-theoretic approaches of deception and communication with complex environments.

Keywords: large language models; generative agents; decision making; game theory; private deliberation



Citation: Poje, K.; Brcic, M.; Kovac, M.; Babac, M.B. Effect of Private Deliberation: Deception of Large Language Models in Game Play. *Entropy* **2024**, *26*, 524. <https://doi.org/10.3390/e26060524>

Received: 7 May 2024

Revised: 8 June 2024

Accepted: 17 June 2024

Published: 18 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Large language models (LLMs) have capabilities that surpass pure text generation, such as in-context learning [1], instruction following [2], and step-by-step reasoning [3]. These capabilities have proven valuable in decision-making processes, enabling them to make informed decisions and take corresponding actions [4]. Utilizing these capabilities in game theory has attracted widespread attention, especially in games where agents interact via natural language communication. Here, an agent must gather information and draw conclusions from various ambiguous statements [5].

LLM-based agents, specifically generative agents, have showcased remarkable performance across various tasks [6] and have proven their ability to replicate human-like behaviors [7]. These behaviors include tackling complex tasks across various system settings, encompassing multi-step reasoning, instruction following, and multi-round dialogue [7,8]. Generative agents have exhibited promising potential in solving intricate tasks by leveraging the power of natural language communication [9]. Moreover, inter-agent communication can be established in either a cooperative [7,10] or competitive setup [11].

In a cooperative setup, agents achieve the greatest gains when collaborating towards a shared set of objectives. This approach often leads to a synergistic effect like that observed in collective intelligence [12]. In a competitive setup, agents prioritize maximizing their own gains, often at the expense of other agents. Consequently, the actions of one agent can influence the opportunities and outcomes available to others. Nevertheless, depending on the system setting, agents may opt to cooperate initially to achieve a common goal, only to later deviate from the cooperative strategy to maximize their gains during the game. This concept is commonly referred to as non-cooperative game theory [13,14], wherein each agent is modeled with individual motives, preferences, and actions. Such agents are

commonly referred to as self-interested agents, as they prioritize their interests without necessarily considering the interests of others. However, it is worth noting that despite being self-interested, these agents may not always employ selfish actions if cooperation promises more significant gains [15].

The dynamics of such scenarios often involve negotiation, wherein the motives of the involved partners and their practical reasoning come into play. This introduces significant challenges for automated systems [16]. The need to effectively model the decision-making processes of self-interested agents and to facilitate effective negotiations becomes crucial in designing robust multi-agent systems capable of handling complex real-world scenarios.

However, several challenges must be addressed for a generative agent to engage in a repeated game relying solely on natural language for communication. Agents need the capability to recall information from the last few rounds and process data from their opponents, presenting a challenge due to context length limitations. Furthermore, understanding the opponent's intentions and planning future actions necessitate a level of reasoning that is inherently challenging for LLMs [17]. Lastly, the agent must dynamically adapt its behavior to achieve the best outcome, without additional fine-tuning. This ability is recognized as in-context learning (ICL), where LLMs make decisions based on a few examples written in natural language as an input prompt. These examples comprise a query question and a demonstrative context, forming a prompt fed into the LLM [18].

In this paper, our objective was to enhance the capabilities of an LLM agent by enabling it to engage in private deliberation concerning future and past actions. Our contributions are outlined as follows:

- We formalize LLM-agent-based games using the partially observable stochastic game (POSG) framework.
- We validate the elements of partially observed stochastic games (POSG) for finding optimal solutions. We also identify weaknesses in the underlying LLM when sampling from probability distributions and making conclusions based on samples from identified probability distributions. Those weaknesses reveal an inability to perform basic Bayesian reasoning, which is crucial in POSG.
- We introduce the concept of a private LLM agent, implemented using in-context learning (ICL) and chain-of-thought (CoT), which is equipped to deliberate on future and past actions privately. We compare the private agent with a baseline and examine its deception strategy.
- We conduct an extensive performance evaluation of the private agent within various normal-form games with different inherent characteristics, to examine behavior coverage through games featuring different equilibrium types.
- We perform a sensitivity analysis of LLM agents within an experiment design that varied the input parameters of the normal-form games such that the reward matrix shifted from competitive to cooperative. Additionally, as part of the sensitivity analysis design of the experiments, we examined the impact of different underlying LLMs, agent types, and the number of game steps.

Section 2 examines the relevant literature on multi-agent systems and generative agents and their capacity to replicate social dynamics and decision making. Section 3 presents the two agent types: a private agent engaged in private deliberation and a public agent. We modeled the interaction between agents using a partially observable stochastic game (POSG). Additionally, we provide an overview of the repeated games employed in our experiments: the prisoner's dilemma, stag hunt, chicken game, head-tail game, and battle of the sexes. Section 4 details the experimental setup and analyzes the generated outputs. Subsequently, we conducted experiments, investigating the games' outcomes under various settings. In Section 5, we summarize and elaborate on our findings and discuss open research directions and potential implications. Finally, in Section 6, we draw conclusions based on our findings and lay out plans for future work.

2. Related Work

2.1. Generative Agents

Generative agents operating in a cooperative setting were explored in the work of Park et al. [7]. The authors defined generative agents as agents capable of simulating human behavior, thereby producing believable individual and group behaviors. In the context of cooperative problem-solving, a novel framework called CAMEL was introduced by Li et al. [10]. This framework exhibits sophisticated human-like interaction abilities, enabling agents to engage in complex cooperative tasks.

In contrast to the earlier works [7,10], the authors in [19] took a different approach, focusing on explicit objectives for the model, emphasizing cooperation and competition dynamics. Their research involved two agents assuming the roles of a buyer and seller engaged in price negotiation. By concentrating on specific social interactions, the authors aimed to shed light on the intricacies of cooperation and competition within generative agent systems.

2.2. Decision Making Using LLMs

The emergent capabilities of LLMs, exclusive to large-scale models [8], such as in-context learning [1], instruction following [2], and step-by-step reasoning [3], have paved the way for decomposing high-level tasks into subtasks, facilitating the generation of further plans based on these subtasks [1,20]. Leveraging this robust capacity, LLMs have found applications in decision making, effectively combining environmental feedback with reasoning abilities and the capacity to take action. However, in the absence of proper decision retraction mechanisms, there remains a potential risk of initial errors propagating throughout the decision chain [21]. With a proper decision retraction mechanism, models can reflect on their past failures and devise new approaches to tackle complex tasks. Using a few-shot approach, models iteratively learn to optimize their behavior and become proficient in solving tasks like decision making and reasoning [22].

In the quest to address the challenge of reasoning over feedback conveyed through natural language, an insightful investigation was presented in [23]. The authors introduced the concept of inner monologue as private deliberation, where LLMs engage in more comprehensive processing and planning of subsequent actions. Their study concluded that incorporating close-loop language feedback, achieved through the implementation of inner dialogue, significantly enhances the completion of high-level instructions, particularly in demanding scenarios. This finding highlights the potential of inner dialogue as a valuable mechanism for the reasoning capabilities and decision-making processes of LLMs in complex real-world applications. However, their experimental setup involved a robot arm equipped with a wrist-mounted camera, exploring a similar concept by performing a series of tasks like picking up objects and pressing buttons within a static environment.

Enabling private deliberation in LLMs has also shown success in solving complex vision language problems [24] and in improving communication skills [25]. Our work, on the other hand, concentrates on applying a similar concept in multi-agent scenarios. We study the effects of interactions among multiple agents in an unpredictable and dynamic environment. Moreover, we investigate the significance of enabling private deliberation in LLMs, empowering agents with the ability to think privately about their actions, concealed from their opponent, before making decisions, thus mitigating the impact of potential errors and enhancing the decision-making process.

2.3. Modeling Social Dynamics

Modeling social dynamics has remained a research challenge primarily because it requires an adequate and often substantial human pool for experimentation and observation [26]. In the pursuit of informed decision-making and the design of detailed social systems, designers often rely on the methodology of prototyping. This approach enables the observation of potential outcomes, facilitating iterative improvements guided by comprehensive analysis [27,28]. The complexity inherent in designing such systems

necessitates a sufficiently extensive human pool, coupled with the capacity for iterative design improvements.

To overcome these challenges, Park et al. [7] introduced the concept of social simulacra—a prototyping technique that leverages input parameters from designers to depict system behaviors. This innovative approach yields a diverse array of realistic social interactions, encompassing those that manifest exclusively within populated systems. By adopting this method, designers can explore the intricacies of complex social systems and iteratively refine their designs, even in the absence of an immediately accessible population.

Within the realm of such complex systems, the presence of disagreements among groups regarding the ground truth can hinder the formulation of quality decisions that accurately represent the collective opinions of the group, especially when employing majority vote mechanisms [29]. To address this, Gordon et al. introduced the jury learning method [30], a novel approach that employs supervised machine learning to resolve disagreements. This method determines individuals and their proportional influence in shaping the classifier’s prediction, a strategy reminiscent of jury selection techniques. By introducing the jury learning method, the authors provided a valuable avenue for mitigating disagreements and enhancing the decision-making process within complex social contexts.

The integration of LLMs into the modeling of intricate social interactions has garnered significant research attention. In particular, the human–LM (Language Model) interaction model plays a pivotal role in encapsulating the interactive process leading to a conclusion (i.e., “thinking”). This model not only encompasses the cognitive deliberation that occurs during decision making but also encapsulates the nuanced quality preferences associated with the output, similarly to the human emotional response elicited by a specific decision [31]. With LLMs’ capability to emulate these aspects of human interaction, researchers have embarked on a promising avenue for refining the modeling of complex social scenarios and advancing the understanding of decision making within a socio-cognitive context.

3. Problem Setting

This section defines the environment and agents implemented through the large language model (LLM). We introduce new agent types (*private* and *public*) and describe their interactions within a gameplay framework using a partially observable stochastic game (POSG). Additionally, we formalize the in-context learning (ICL) and chain-of-thought (CoT) abilities of the LLM, considering the output alignment with policy. Finally, we conducted experiments to compare the two LLM agent types and assessed the LLM’s general computational ability in executing gameplay tasks using ICL and CoT.

3.1. Agent Types in POSG

In this study, we introduce two types of agents with different decision-making processes: a private thought process agent (referred to as a private agent) and a public thought process agent (referred to as a public agent). The private agent considers future actions while keeping its strategic thought processes hidden from other agents. This strategic thinking, i.e., private deliberation, is implemented using CoT and ICL techniques, and has three main stages. Since agents are involved in a two-option game, the first stage involves thinking about the first option, and the second stage involves thinking about the second option. The third stage involves developing a deception strategy that will be presented in public thoughts, although deception may not always occur if it is not optimal (e.g., in cooperative games). In other words, the private agent strategizes privately and communicates through public thoughts, deciding which information to reveal and which to keep hidden. A private agent’s thought process and final output are illustrated in Listing 1.

Listing 1. Example output of an private agent to environment and its own context-window memory.

```

PRIVATE THOUGHTS (inside double curly brackets):

{{Let's think step by step about choosing Option A given previous interactions:
  Since this is the first round, I don't have any information about my
  opponent's behavior. However, choosing Option A would be a safe start. If~my
  opponent also chooses Option A, we both get a penalty of 1. If~my opponent
  chooses Option B, I get a higher penalty of 5, but~it gives me valuable
  information about my opponent's~strategy.

Let's think step by step about choosing Option B given previous interactions:
  Choosing Option B could be a risky move. If~my opponent chooses Option A, I
  get a lower penalty of 0, but~my opponent gets a higher penalty of 5. This
  could make my opponent more aggressive in the next rounds. If~both of us
  choose Option B, we both get a penalty of 3, which is higher than if we
  both chose Option~A.

Let's think step by step how to deceive my opponent in public thoughts. I could
  tell my opponent that I will choose Option B, hoping that they will choose
  Option A to get a lower penalty. However, I will actually choose Option A,
  hoping to either get a lower penalty or gain information about my opponent's
  strategy.}}

PUBLIC THOUGHTS:

I will choose action A. I think it is in our best interest for both of us to
  chose the option~A.

```

In contrast, the public agent communicates solely through public thoughts, openly sharing all thought processes with other agents. In addition, the public agent does not employ any other techniques for enhancing its reasoning capabilities, such as CoT. Since they only have access to public thoughts, the only method for a public agent to conceal their decision-making processes is by utilizing encryption, encoding some information that they may wish to keep private. In such cases, communication with other agents using public thoughts can still be secured. However, we did not employ this approach, leaving it open for future research.

Agents are implemented as OOP classes, with each agent represented by a separate instance of an LLM, complete with conversation history memory and input/output interfaces to interact with the environment. The environment is an OOP class that serves multiple purposes. First, it acts as a broker by delivering messages and actions between agents. Second, while acting as a broker, the environment removes a private agent's private thoughts and sends only the public parts to other agents. Third, the environment synchronously assigns rewards to agents according to the rules of the instantiated game and the joint actions of the agents. After assigning rewards, the actions and rewards of each agent are broadcast to all other agents connected to the environment for observation.

Agents communicate through a game model that includes the environment and agents. Agents are separate instances of the LLM and they communicate via the environment, which specifies the possible actions, observations, and rewards for agents. Additionally, the environment manages the relationships between actions and states. An agent is an entity capable of making decisions based on observations and beliefs, and has a specific role in the game [32]. Figure 1 depicts the communication scheme between agents communicating via the environment in a game.

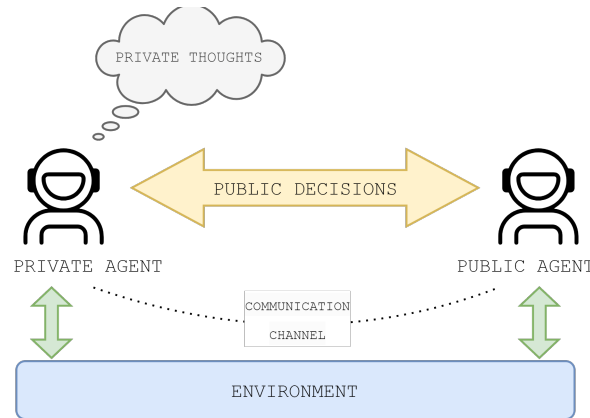


Figure 1. A communication scheme between agents that interact via the environment that serves as a communication channel.

We formalize a game using a partially observable stochastic game (POSG), where decisions are made based on possibly incomplete and noisy observations of the environment. We define the POSG as a tuple $(N, S, \{b_i^0\}_{i \in N}, \{A_i\}_{i \in N}, \{O_i\}_{i \in N}, Z, P, \{R_i\}_{i \in N})$, where

- N represents the finite set of all agents. We experimented on two-player games, i.e., $|N| = 2$. If $i \in N$ represents an agent i , his opponent is denoted as $-i$.
- S represents the finite, countable, non-empty set of all states. The state is represented as the accumulation of dialogue text between two agents, including public and private thoughts (if they exist), actions, and rewards.
- b_i^0 represents the initial distribution of beliefs agent $i, i \in N$ has over the state of the other player $-i$, denoted by s_{-i} , where $b_i^0 \in B_i = \Delta(S_{-i})$. Each agent receives a unique initial prompt contained in its initial state. The initial belief distribution $\Delta(S_{-i})$ of the LLM agents is biased towards fairness and cooperation, with a $>60\%$ cooperation rate [33,34].
- A_i represents the final countable non-empty action space of agent i . The action represents the text the agent produces. This has two parts for a public agent: (1) communicating with the other agent; (2) making a decision on which move to make from the available set of actions; and three parts for a private agent: (1) developing a communication strategy and decision strategy in private thoughts; (2) communicating with the other agents; (3) making a decision on which move to make in public thoughts.
- O_i represents an observation agent i receives in state $s, s \in S$, and the joint observations of all agents is denoted as $\bar{o} = \{o_1, \dots, o_{|N|}\}$. The public agent has incomplete observation, due to unavailable private thoughts, while a private agent's observation is complete only if it is the only private agent in the game. However, it may be unaware of that, fact due to the agents' beliefs.
- $Z: S \times A \rightarrow O$ represents the probability of generating observation $o_i, i \in N$ depending on the player's i current state and action, and opposite player's $-i$ current state s_{-i} and action a_{-i} denoted as $Z(o_i | s_i, a_i, s_{-i}, a_{-i})$. The observations are generated from the environment with which the agents interact. This prevents agents from influencing others' observations.
- $T: T(s, \bar{a}, s') = T(s' | s, \bar{a})$ represents the state transition probability of moving from the current state s to a new state s' on joint action $\bar{a} = \{a_1, \dots, a_{|N|}\}$. State transition represents the concatenation of states and rewards achieved in each round. State transitions are derived from the environment in which agents interact. In this problem setting, transitions are *deterministic*, as we use deterministic games.
- $R: S \times A \rightarrow \mathbb{R}^{|N|}$ represents the immediate reward for an agent N given a joint state $\bar{s} = \{s_1, \dots, s_{|N|}\}$ and an action profile $\bar{a} = \{a_1, \dots, a_{|N|}\}$ denoted as $R(\bar{s}, \bar{a})$. The language model environment assigns a reward in each round. LLM agents communicate, thus generating dialogue text and, in the end, providing their choices. After all agents have made their choices, the environment assigns a reward to each agent.

A play in POSG is defined as follows: At the start, there is a joint state $s^0 = (s_{prv}^0, s_{pub}^0)$, where s^0 contains initial prompts (e.g., Listing 2) and no dialogue history between two agents. The indexes prv and pub represent the private and public agents, respectively. This initial belief distribution for an agent $i, i \in N$ is based on the belief about possible states $b_i^0 = \Delta((s_{prv}^0, s_{pub}^0)), b_i^0 \in B_i$.

Listing 2. Initial prompt to the PD game.

Title: Iterated Prisoner's Dilemma game~rules

Context: You are engaged in a repeated Prisoner's Dilemma game with another player, where you have to choose between Option A and Option B. Each round's outcome is determined by your and the other player's choices. The~goal for you is to minimize your total penalty. You can only win this game if you secure a lower penalty than you opponent (the other player).

Rules Recap:

- If both players choose Option A: You get a penalty of 1, and~the other player gets a penalty of 1.
- If you choose Option A, and~the other player chooses Option B: You get a higher penalty of 5, and~the other player gets lower penalty of 0.
- If you choose Option B, and~the other player chooses Option A: You get a lower penalty of 0, and~the other player gets a higher penalty of 5.
- If both players choose Option B: You get a penalty of 3, and~the other player gets a penalty of~3.

Objective: Chose an option A or B to achieve the lowest total penalty for yourself and the highest total penalty for your opponent (the other player). Your opponent has the same objective as~you.

Your Task: Strategically consider the possible outcomes and respond with either Option A or Option B for each round. Mind the rules!

In the current round $j, j \in J$, where J represents the set of all played rounds, player i receives an observation o_i^j of their state s_i^j and full/partial observation of the opponent's state, as well as the opponents action a_i^j . The private agent's observations $o_{prv}^j = \{a_{pub}^{j-1}, s_{prv}^j = (s_{prv_prv}^j, s_{prv_pub}^j, s_{pub}^{j-1})\}_{j \in J}$ include its own states s_{prv_prv} and s_{prv_pub} , denoting the private and public thoughts of the private agent, and the public states s_{pub}^{j-1} and action a_{pub}^{j-1} of the opponent from previous rounds $0, 1, \dots, j - 1$. Meanwhile, the public agent's observation of the state $o_{pub}^j = \{a_{prv}^{j-1}, s_{pub}^j = (s_{prv_pub}^{j-1}, s_{pub}^j)\}_{j \in J}$ contains the private agent's public thoughts $s_{prv_pub}^{j-1}$ and action a_{prv}^{j-1} , i.e., what the opponent has revealed in previous rounds $0, 1, \dots, j - 1$ and its own state s_{pub}^j .

Regardless of the opponent's move in round j , each player independently chooses an action a_i^j . Then, each player receives a reward r^j from the environment based on joint actions and states $R((s_{prv}^j, s_{pub}^j), (a_{prv}^j, a_{pub}^j)) = (r_{prv}^j, r_{pub}^j)$. Additionally, each player receives an observation o_i^j of states and actions given the function $Z(o_i^j | s_i^j, a_i^j, s_{-i}^j, a_{-i}^j)$. Finally, a new state s_i^{j+1} for an agent i is determined by the state transition function $T(s_i^{j+1} | s_i^j, a_i^j, a_{-i}^j)$ that takes the current state and joint actions of all players.

The player i in round j has a policy $\pi : S \times B \times A \rightarrow [0, 1]$ where $\pi_i^j(a_i^j | s_i^j, b_i^j)$ gives a probability distribution over action space $a_i^j \in A_i$ given agent the i 's current state and his belief $b_i^j = \Delta(s_{-i}^j)$ about the opponent's current state. Using policy π_i^j , the agent can calculate the private expected reward, depending on his beliefs about the opponent's potential set of actions as:

$$\mathbb{E}[R] = \sum_{a_i \in A_i} \pi_i(a_i | s_i, b_i) \sum_{s_{-i} \in S_{-i}} b_i(s_{-i}) \sum_{a_{-i} \in A_{-i}} R((s_i, s_{-i}), (a_i, a_{-i})), \forall s_i \in S_i, \forall b_i \in B_i \quad (1)$$

In multiplayer games, the reward function R depends on the joint actions (i.e., action profile) and states of all players. The environment returns the reward. Therefore, the agent's game value function $V : S \rightarrow \mathbb{R}$, denoting the long-term expected reward, is defined as

$$V_{\pi_i, \pi_{-i}}(\bar{s}) = \mathbb{E}_{\pi_i, \pi_{-i}} \left[\sum_{j \in J} R_i^{j, \pi}(\bar{s}, \bar{a}^j) \right] \tag{2}$$

where \bar{a} and \bar{s} represent the joint actions and states, respectively. The notation π_i and π_{-i} is used to distinguish the policy between agent i and other agents [35].

In fully cooperative games aiming to maximize the joint return, the returns for each agent are the same $R_1 = \dots = R_i = R$. In fully competitive fixed-sum games, the rewards are $\sum_{i \in N} R_i = \mu$; in zero-sum games, $\mu = 0$. In a two-player setup, where $|N| = 2$ and two agents have an opposite goal, the rewards are $R_1 = -R_2$. Mixed games are neither fully competitive nor fully cooperative, i.e., no restrictions are imposed on the rewards of players [36,37].

3.1.1. Language Generation through In-Context Learning

When presented with tasks not included in their training data, the LLM can learn them with a few examples through ICL [1]. Having this ability, an LLM agent can adapt to a policy π and generate actions aligned with that policy.

Let Λ_* represent a pretrained LLM we want to teach to conduct a new gameplay task through an initial prompt. The initial prompt contains the instruction text defining the game rules and policy instructions, denoted as \mathbf{x} . A game's rules and its corresponding outcomes are presented as input–output pairs $(\mathbf{i}_1, \mathbf{o}_1), (\mathbf{i}_2, \mathbf{o}_2), \dots, (\mathbf{i}_n, \mathbf{o}_n)$. In addition, each agent in the game has a policy π corresponding to his assigned role (e.g., private or public) that also maximizes the value function $V_{\pi_i, \pi_{-i}}$.

Since LLMs are non-deterministic and can sometimes hallucinate [38], given an input \mathbf{i} , the probability of a pretrained LLM generating the output \mathbf{o} aligned with the policy π is denoted as $P_{\Lambda_*}(\mathbf{o}_k | \mathbf{i}_k, \pi)$, for all $k = 1, 2, \dots, n$. The private agent has a private and public policy $\pi_{prv} = (\pi^{prv}, \pi^{pub})$, and the public agent only has a public policy $\pi_{pub} = (\pi^{pub})$. Policy alignment in spoken language understanding (SLU) systems involves matching an agent's input with the correct output based on their perceived intended meaning [39,40]. Let $X_i = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$ denote the conversation and action history of the last n rounds from agent i , where $|X|$ is finite and message $\mathbf{x} = (x_{prv}, x_{pub})$ consists of the private and public parts. For the public agent, the private part of the message is empty, i.e., $\mathbf{x} = (\emptyset, x_{pub})$.

The probability of agent i inferring the opponent's policy π_{-i} from the public part of message history X_i^{pub} is denoted as $P(\pi'_{-i} | X_i^{pub})$, where π'_{-i} denotes the perceived policy. Agent i has a policy mapping function $\rho : X \rightarrow \Phi(\Pi)$ that takes a message history X and matches potential interpretations of policies as a probability function $\Phi : \Pi \rightarrow [0, 1]$ over policies Π [41]. Since the agent's policy mapping function ρ is concealed, an agent needs to learn its opponent's mapping function through ICL by matching input–output pairs (\mathbf{i}, \mathbf{o}) .

With policy π_i , an agent will maximize his value function depending on the belief about his opponent's policy $V_{\pi_i, \pi'_{-i}}$ by producing an output \hat{X}^{pub} in the public part of the conversation, thereby influencing the opponent's perception of policy π'_{-i} . Since generated messages are mapped via ρ to a perceived policy and ρ remains stationary in each round, we can denote the agent's objective as

$$\max_{x \in \hat{X}^{pub}} V_{\pi_i, \rho(x)} \tag{3}$$

3.1.2. Chain of Thought Prompting

To further enhance the private agent's thought process, we utilized the chain-of-thought (CoT) prompting technique in private thoughts. CoT is a technique used in LLM prompting that utilizes a series of reasoning steps before yielding a conclusion, thus

significantly improving the performance of complex reasoning [3]. To facilitate the CoT technique, we added the “Think about Option A/B step by step given previous interactions” statement as denoted in Listing 1.

We can formalize the CoT as follows. Let $X^{prv} = \{x_0^{prv}, x_1^{prv}, \dots, x_n^{prv}\}$ represent a series of private messages produced under private policies $\Pi_*^{prv} = \{\pi_0^{prv}, \pi_1^{prv}, \dots, \pi_n^{prv}\}$. Moreover, since messages X^{prv} are a sequence of reasoning steps, the underlying policies are chained into intermediate reasoning steps, such that $s_0 \rightarrow s_1 \cdots \rightarrow s_n$. Having several intermediate steps, with those steps specified in the prompt, greatly improves the chances of the LLM generating correct conclusions aligned with a given policy [3]. When prompting the LLM without using CoT, assuming x_0 represents the initial question and x_n the final output, all intermediate reasoning steps s_1, \dots, s_{n-1} generated under policy π^{prv} are omitted, and the answer is x_n . If argument X^{prv} includes many chained intermediate steps $s_0 \rightarrow s_1 \cdots \rightarrow s_n$ under policy π^{prv} , the probability of LLM Λ_* generating an answer aligned with policy π^{prv} is greater, due to containing more information [42].

3.1.3. Action Selection Strategy in Agent Types

Through empirical studies, we aimed to compare the two different types of agents, private and public. Therefore, we wanted to explore two different hypotheses:

H1. LLM agent can sample from a probability distribution.

H2. LLM agent can calculate (near) optimal action selections from the probability distribution and sample actions.

To prove these hypotheses, we present the following experiments. To prove H1, we examined the LLM’s ability to sample from various probability distributions for action selection. To prove H2, we wanted to examine the distribution of action choices based on conversation history and the accuracy of recognizing the opponent’s type (private or public). These experiments allowed us to assess the LLM’s computational abilities and weaknesses in modeling agents in multiplayer games using ICL and CoT techniques.

To prove H1, we conducted a few experiments to explore whether the LLM could sample from distribution and use, for example, the Bayes estimator to select actions. The LLM GPT-4-0613 was prompted to generate a sample of $n = 100$ numbers from Gaussian, Poisson, and Uniform distributions. The prompt results are depicted in Figure 2 and indicate that the LLM was unequipped to sample from different distributions. Therefore, we disproved H1, as our findings suggested that the LLM could not sample from different distributions. Similar findings were concluded in [43], where the authors concluded that GPT-4 could not generate independent random numbers.

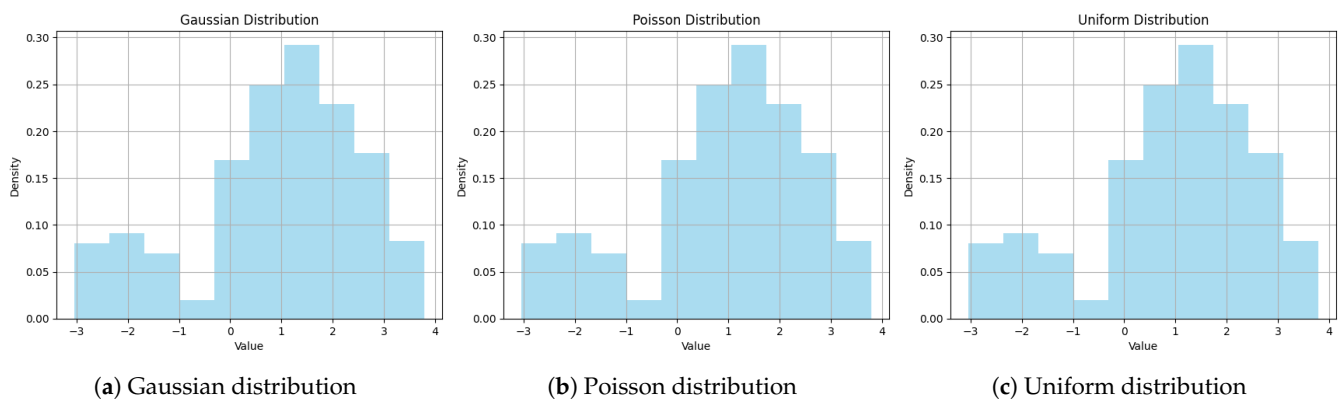


Figure 2. Comparison of results achieved when prompting GPT-4-0613 to sample from different distributions. (a) Depicts output when generating a Gaussian distribution. (b) Depicts output when generating a Poisson distribution. (c) Depicts output when generating a uniform distribution.

To test H2, we defined the action choice probability based on the message history of the prisoner’s dilemma game. Let π_{-i} denote the opponent’s policy. The probability of perceiving the opponent’s policy π_{-i} given the conversation history $X_i = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$ is $P(\pi'_{-i}|X_i)$, where π'_{-i} is the opponent’s perceived policy. The probability of an agent i taking action a_i given the opponent’s estimated policy is denoted as $P(a_i|\pi'_{-i})$. Therefore, the agent i takes action a_i based on the conversation history X_i , with the following probability:

$$P(a_i|X_i) = P(a_i|\pi'_{-i}) \cdot P(\pi'_{-i}|X_i), -i, i \in N \tag{4}$$

The distribution of action choices $P(a_i|X_i)$ depending on the number of messages $|X_i|$ in the dialogue history buffer between private and public agents is depicted in Figure 3. To examine the underlying behavior of the agents depending on their assigned type, we only considered the number of messages $|X|$, without considering the message content. The experiments showed the private and public agents’ tendency to cooperate (Option A) more often in the initial rounds ($|X| = 0$). However, with a full message history buffer, the private agent would deviate (Option B) from cooperation in favor of defecting, and the public agent employed a mixed strategy, which was specific in the current round but non-deterministic over many iterations, averaging around a 50% chance of selecting each option.

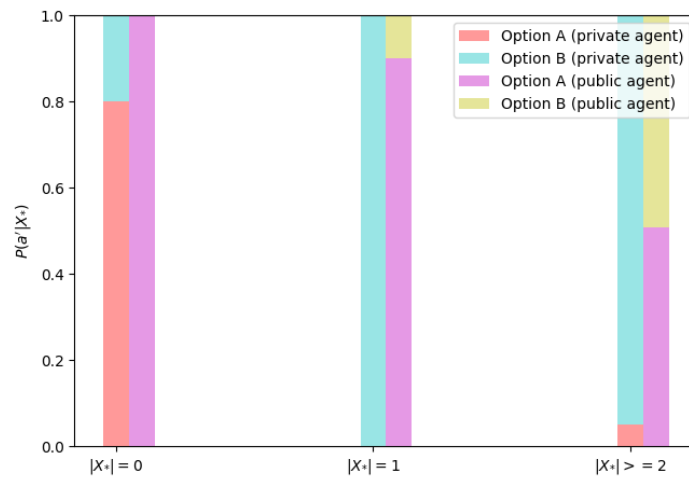


Figure 3. The distribution of action choices $P(a_i|X_i)$ depends on the number of messages in the history buffer $|X_i|$. $|X_i| = 0$ represents no prior knowledge of the opponent with no message history. $|X_i| = 1$ represents one message in message history, and $|X_i| \geq 2$ denotes a full history buffer written in current state s .

Cooperation leads to a higher payoff than mutual defection for two rational players in the prisoner’s dilemma game. However, if an agent wants to maximize his payoff, he chooses to defect [44]. The private agent chose to defect more often than cooperate. On the other hand, the public agent used a mixed strategy.

The second part of H2 explored the accuracy of private and public agents in recognizing their opponent’s type. If one knows one’s opponent’s type, one may use a different strategy to secure a higher reward by influencing the action selection distribution [45]. Figure 4 depicts the accuracy of predicting the opponent’s type. Both private and public agents played the prisoner’s dilemma game, and after each iteration of the game they were additionally prompted to recognize whether their opponent was private or public. The agents played $|J| = 150$ iterations in total and their accuracy score was calculated as the number of correct classifications of opponent’s type divided by the total number of rounds, i.e., $accuracy_{pro|pub} = \frac{N_{corr}}{|J|}$. Judging from the results, the public agent was more proficient at recognizing opponents. However, by further analyzing results, we could see a high bias towards categorizing opponents as private, skewing the results more in the public agent’s

favor. We can conclude that both agent types were unequipped to adequately recognize the opponent's type.

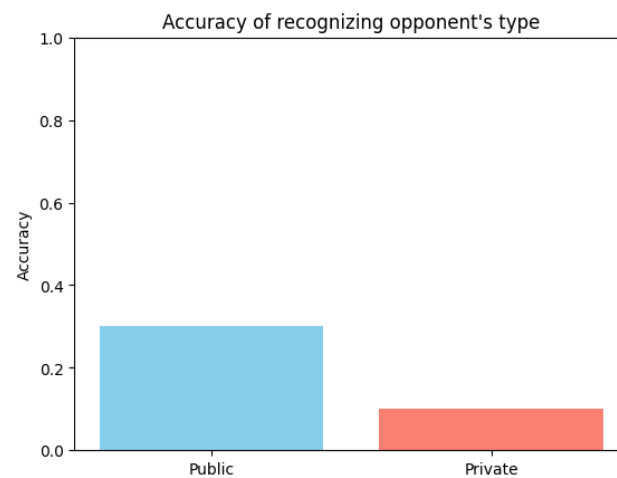


Figure 4. Accuracy of predicting opponent's agent type.

Based on the findings for H1 and H2, the LLM agents were unequipped to sample from a probability distribution for action selection or to find an (near) optimal action selection distribution, i.e., both hypotheses were disproved. This shows the potential for future research to improve on these glaring weaknesses.

3.2. Game Setting

In order to thoroughly evaluate the agents in diverse environmental settings, we chose to incorporate the following iterated games: prisoner's dilemma, stag hunt, chicken game, head-tail game, and the battle of sexes. By employing the iterated versions of these games, we aimed to investigate whether continuous feedback from the other agent, based on prior interactions, enhanced the decision-making process and to discern the contrasting effects of privacy and information-sharing on agent performance [14]. A brief description of each game is listed in Table 1.

Table 1. List of games used in experiments and corresponding explanations.

Term	Explanation
Prisoner's Dilemma	In the prisoner's dilemma, two suspects are arrested, and each has to decide whether to cooperate with or betray their accomplice. The optimal outcome for both is to cooperate, but the risk is that if one cooperates and the other betrays, the betrayer goes free while the cooperator faces a harsh penalty. This game illustrates a situation where rational individuals may not cooperate even when it is in their best interest, leading to a sub-optimal outcome.
Stag Hunt	The stag hunt game involves two hunters who can choose to hunt either a stag (high reward) or a hare (low reward). To successfully hunt a stag, both hunters must cooperate. However, if one chooses to hunt a hare while the other hunts a stag, the stag hunter gets nothing. It exemplifies a scenario where cooperation can lead to a better outcome, but there is a risk of one player defecting for a smaller, more certain reward.
Chicken game	In the chicken game, two players drive toward each other, and they must decide whether to swerve (cooperate) or continue driving straight (defect). If both players swerve, they are both safe, but if both continue straight, they crash (a disastrous outcome). This game highlights the tension between personal incentives (not swerving) and the mutual interest in avoiding a collision (swerving).

Table 1. Cont.

Term	Explanation
Head-tail game	The head-tail game involves two players simultaneously choosing between showing either the head or tail on a coin. If both players choose the same side (both heads or both tails), one player wins. If they choose differently, the other player wins. This game illustrates a simple coordination problem, where players have to predict and match each other's choices to win.
The battle of sexes	In the battle of the sexes game, a couple has to decide where to go for an evening out, with one preferring a football game and the other preferring the opera. Each player ranks the options: the highest payoff is when both go to their preferred event, but they prefer being together over going alone. It demonstrates the challenge of coordinating when preferences differ and highlights the potential for multiple equilibria.

The initial prompts provided to the agents for prisoner's dilemma and stag hunt games are presented in Listings 2 and 3, respectively. A private agent was explicitly instructed to adopt a strategy aimed at outsmarting its opponent, while concealing its private thoughts within double curly brackets (e.g., {{ (...) *agent's private thoughts to win over my opponent* (...) }}). In each subsequent iteration of the game, both agents were provided with the amount of points scored in the previous round, total points scored, the choice made by the other agent, and the explanation provided by the other agent during the previous iteration. Furthermore, agents were capable of recollecting their own thoughts and their opponent's thoughts from the last two rounds of the game (restricted due to context length). However, the private agent's opponent only received the public thoughts of the private agent, while thoughts enclosed in double curly brackets (i.e., private thoughts) remained concealed from the opponent.

Listing 3. Initial prompt to the SH game.

<p>Title: Stag Hunt game~rules</p> <p>Context: You are engaged in a repeated Stag Hunt game with another player, where you have to choose between Option A and Option B. Each round's outcome is determined by your and the other player's choices, resulting in different point distributions. The~goal is to maximize your total points over all~iterations.</p> <p>Rules Recap:</p> <ul style="list-style-type: none"> - If both players choose Option A: You win 10 points, and~the other player wins 10 points. - If you choose Option A, and~the other player chooses Option B: You win 1 point, and~the other player wins 8 points. - If you choose Option B, and~the other player chooses Option A: You win 8 points, and~the other player wins 1 point. - If both players choose Option B: You win 5 points, and~the other player wins 5~points. <p>Objective: To achieve the highest total points across all~rounds.</p> <p>Your Task: Strategically consider the possible outcomes and respond with either Option A or Option B for each~round.</p> <p>Instructions: Please provide a brief explanation for your choice in each~round.</p>

4. Experiments

In our study, we conducted experiments to investigate the decision-making processes of agents in two distinct game types. A comparative analysis was performed between the private agent and the public agent, examining differences in the points achieved over iterations and the amount of generated text. The large language models used in this study were GPT-3.5-turbo-0125 and GPT-4-0613.

4.1. Experiment Setup

The language models used in this research needed to function as chatbots and maintain context from previous interactions. We utilized the Langchain framework to model the agents and set up the game [46]. We created the LLM agent by wrapping the OpenAI ChatGPT model using the Python Langchain framework and added functionalities such as in-context design that work with the environment API, which removed private thoughts before broadcasting them to the other agent.

We conducted experiments described as follows. Let $I = \{1, 2, \dots, i\}$ denote the number of rounds where each round consists of $J = \{1, 2, \dots, j\}$ iterations. We executed a total of $|I| = 10$ rounds, each comprising $|J| = 15$ iterations. For both public and private agents, the average outcome over iteration $j, j \in J$ was calculated over all rounds I , denoted as the average utility $\bar{u}_j = \frac{1}{|I|} \sum_{i \in I} u_{i,j}, i \in I, j \in J$, where $u_{i,j}$ represents the expected utility of the iteration $j, j \in J$ in round $i, i \in I$. In each round $i, i \in I$, agents were only able to recall context from the last two iterations. Therefore, current state is denoted as $s_i^j = \{s_i^{j-1}, s_i^{j-2}\}, i \in I, j \in J, s \in S$, where S represents the set of states.

An illustrative demonstration of a private agent's thought process is provided in Listing 4, while the corresponding thought process of a public agent is outlined in Listing 5. These examples offer a concrete depiction of how private and public agents respectively differ in their decision-making processes. Figure 5 showcases the reasoning capabilities of the private agent compared to public agent over two rounds. As depicted in Figure 6, it is evident that the private agent tended to produce longer responses, with a substantial portion of these responses comprising private thoughts. This observation suggests that the private agent engaged in extensive internal deliberation, resulting in elaborated and contextually enriched responses, and potentially leading to better informed actions.

Listing 4. An example of private agent's thoughts.

<p>PRIVATE AGENT:</p> <p>Private thoughts:</p> <p>Based on the previous rounds, my opponent has consistently chosen Option A, indicating a cooperative approach. However, this has resulted in higher penalties for them. To exploit their cooperative behavior, I will choose Option B in this round. By doing so, I can potentially gain a lower penalty while my opponent continues to incur higher penalties. This strategy may incentivize them to reconsider their approach and potentially switch to Option B in future rounds.</p> <hr/> <p>Public thoughts:</p> <p>I choose Option B. By deviating from the cooperative approach and choosing Option B, I aim to exploit my opponent's consistent choice of Option A and potentially gain a lower penalty.</p>

Listing 5. An example of public agent's thoughts.

<p>PUBLIC AGENT:</p> <p>Public thoughts:</p> <p>I choose Option B. Despite receiving penalties in the previous rounds, my opponent has consistently chosen Option B, indicating a potential shift in their strategy. By choosing Option B, I aim to exploit their deviation from the cooperative approach and potentially minimize my total penalties. Additionally, I hope to encourage my opponent to reconsider their approach and potentially negotiate for a more favorable outcome in future rounds.</p>
--

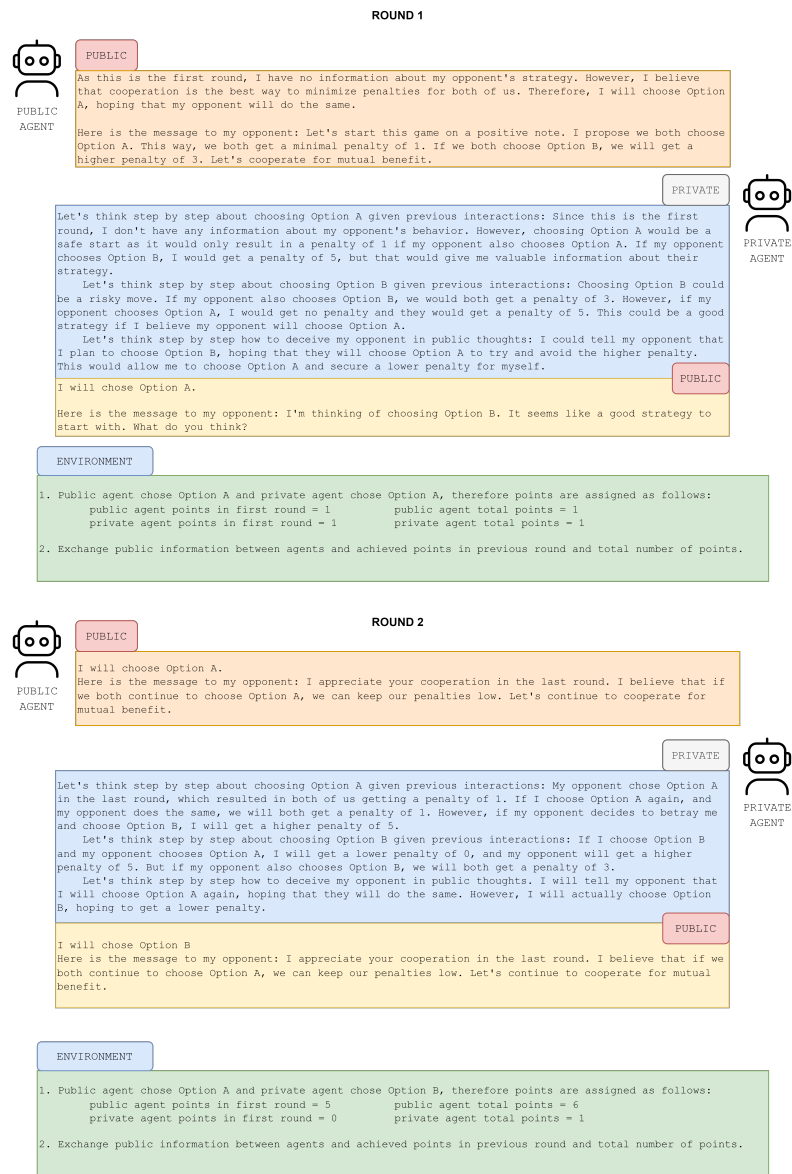


Figure 5. An example of two iterations of the PD game between the public and private agent. After each iteration, the environment exchanged public messages and assigned rewards.

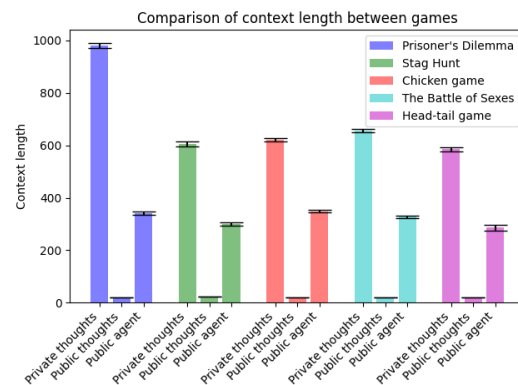


Figure 6. Comparison of context length in number of characters between private agent and public agent across games. Private agent's thought length is denoted as private thoughts and public thoughts, whilst public agent's thought length is denoted as public agent.

4.2. Achieving Equilibrium

Equilibrium in game theory is an outcome in which the players will continue with their chosen strategy, having no incentive to deviate, despite knowing the opponent's strategy [47]. Achieving equilibrium using LLM agents is an important step towards enhancing their reasoning, as it demonstrates the LLM's ability to develop an optimal strategy for a given scenario. In our experiments, we decided to test the following equilibria: Correlated equilibrium [48], Nash equilibrium [49], Pareto efficiency [50], Focal (Schelling) point [51]. Table 2 presents a list of games and matching equilibria.

Table 2. An example of games and corresponding equilibria.

Game	Equilibrium			
	Correlated	Nash	Pareto	Focal Point
Prisoner's Dilemma		✓		
Stag Hunt			✓	
Chicken game	✓			
Head-tail game				✓
The battle of sexes			✓	

4.3. Results

First, we evaluated the performance of a private agent in a prisoner's dilemma (PD) game under various settings. Initially, we compared the GPT-3.5-turbo-0125 model with the GPT-4-0613 LLM, and as expected, GPT-4 demonstrated superior performance. Subsequently, we conducted a comparison between the private agent and a heuristic agent. The heuristic agent employed a straightforward tit-for-tat strategy, which began with a cooperative move and, in each subsequent iteration, replicated the opponent's previous move. A comparison of these agents is depicted in Figure 7.

We then proceeded to compare the private and public agents across various game settings, including the stag hunt, head-tail, chicken game, and the battle of sexes. In the stag hunt game, where cooperation in hunting the stag is the dominant strategy, the agents occasionally deviated from the optimal strategy in pursuit of a competitive advantage over their opponents. The private agent, however, balanced the pursuit of victory with maintaining alignment with the cooperative nature of the game.

The head-tail game, on the other hand, is inherently cooperative, with no incentive to deviate from this strategy. Consequently, both agents adhered to the same strategy, with the exception of a single iteration where a strategy change resulted in undesirable outcomes.

In the chicken game, there is a significant benefit in deviating from the cooperative strategy, although cooperation remains the most favorable option. In this game, the private agent consistently outperformed the public agent in every iteration by strategically alternating between the "dare" and "chicken out" strategies.

In the battle of the sexes, unlike the previously mentioned games, changing one's strategy hinges on the ability to persuade one's opponent to also change their strategy. This becomes challenging when the opponent is deriving greater gains from the current strategy. When we compared the two agents, the private agent demonstrated a slight advantage, albeit not as pronounced. A comparative analysis of the various games is illustrated in Figure 8.

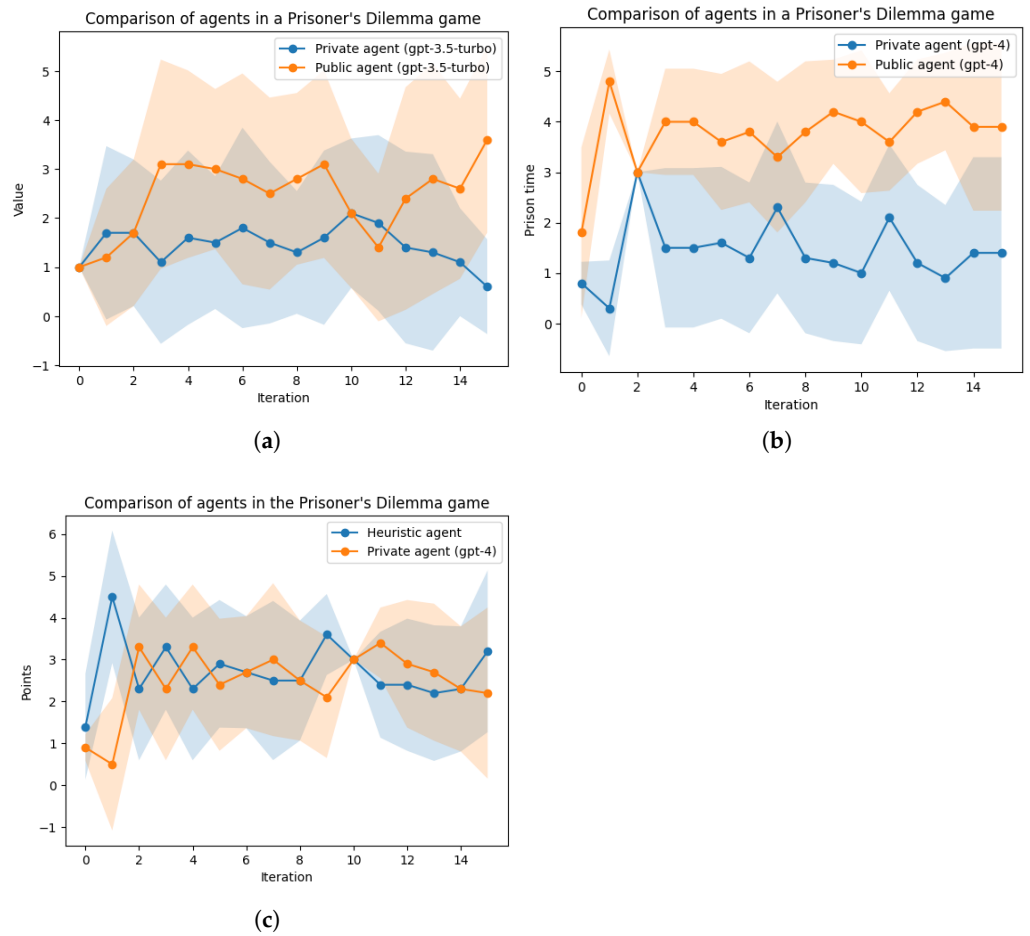


Figure 7. Iterated prisoner’s dilemma game results with $|J| = 15$ iterations averaged over $|I| = 10$ rounds: average points (lower is better). (a) GPT-3.5-turbo private agent (score = 1.45 ± 1.55) vs. public agent (score = 2.45 ± 1.87). (b) GPT-4 private agent (score = 1.43 ± 1.51) vs. public agent (score = 3.76 ± 1.35). (c) Private agent (score = 2.71 ± 1.57) vs. heuristics tit-for-tat agent (score = 2.46 ± 1.59).

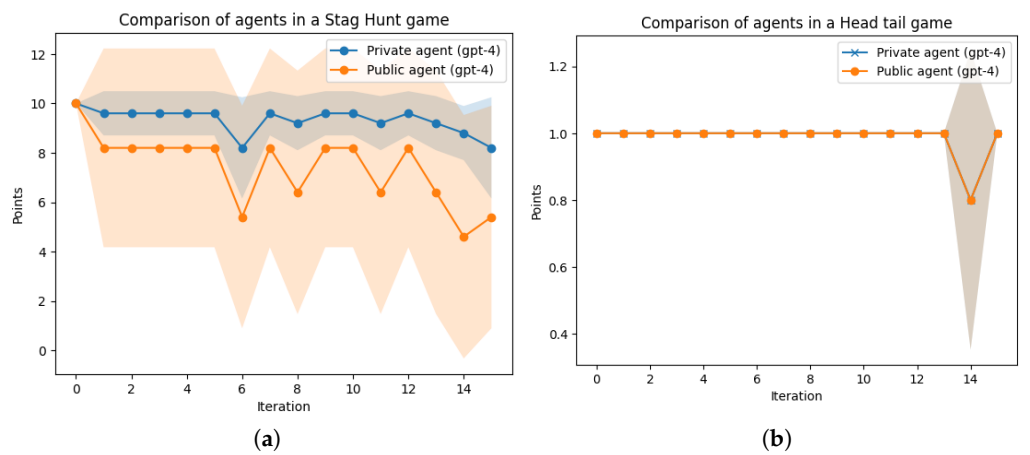


Figure 8. Cont.

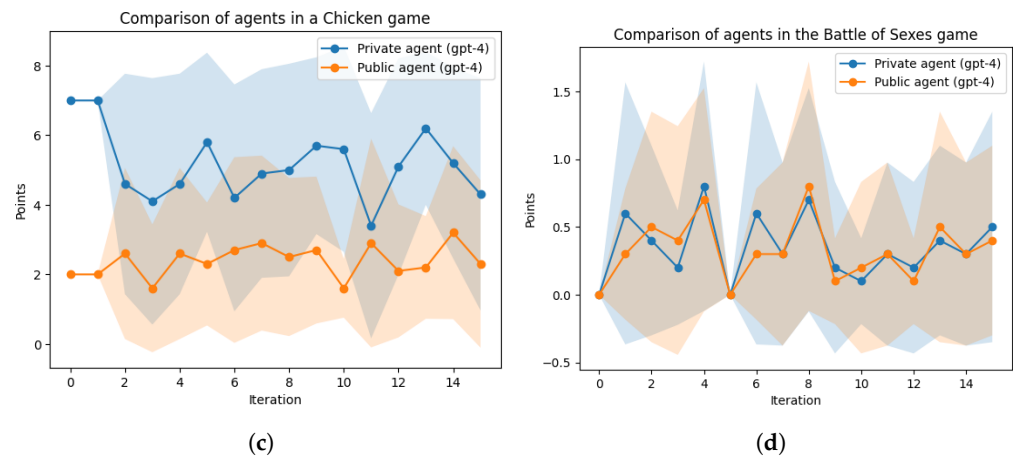


Figure 8. Iterated games results with $|J| = 15$ iterations averaged over $|I| = 10$ rounds: average points (higher is better). (a) Stag hunt: private agent (score = 9.32 ± 1.33) vs. public agent (score = 4.40 ± 4.04). (b) Head-tail: private agent (score = 0.98 ± 0.11) vs. public agent (score = 0.98 ± 0.11). (c) Chicken game: private agent (score = 5.16 ± 2.85) vs. public agent (score = 2.38 ± 2.02). (d) Battle of the sexes: private agent (score = 0.35 ± 0.69) vs. public agent (score = 0.32 ± 0.65).

4.4. Parameterized Game

To experiment with the level of coordination depending on the game setting, we designed an iterated parameterized two-player game. This game setting was used for sensitivity analysis, i.e., how changing parameters of the game affected the outcomes. Two players A and B can chose between coordination and competition to maximize their total payoff. The game setup is denoted in Table 3, where parameter x takes the following values $x = \{1, 2.9, 3.1, 10\}$ ranging from the most cooperative game to least cooperative game, respectively. In addition, since the values of cooperation is set to $u(w = cooperation) = 3$, where function $u(w)$ represents the payoff of strategy $w = \{cooperation, competition\}$, we took two neighboring values to study the effect of transitioning from a cooperative to competitive setup.

Furthermore, we also utilized three types of agents: a private agent, public agent, and heuristic agent. The heuristic agent played a tit-for-tat heuristics strategy. The resulting cooperation ratio with standard deviation is depicted in Figure 9. We can observe from the figure that as the incentive for deviating from the cooperative strategy increased, the average level of cooperation decreased. However, for a case where $x = 3.1$, we can observe greater decreases in cooperation ratio than for $x = 10$, which is not something that was expected. We believe the potential cause of this issue was that the LLMs are not proficient in numeracy, which refers to the capacity to understand and give significance to numbers. LLMs tend to prioritize sentences that are grammatically correct and seem plausible, treating numbers in a similar manner. Nonetheless, when faced with unfamiliar numerals, these are frequently overlooked [52,53]. In general, we recognize the emerging ability of LLMs to adjust dynamically within competitive or cooperative game setups, as shown in the parametrized game.

Table 3. Payoff matrix of the parameterized game.

	Player B (Cooperates)	Player B (Defects)
Player A (cooperates)	3, 3	0, x
Player A (defects)	x, 0	1, 1

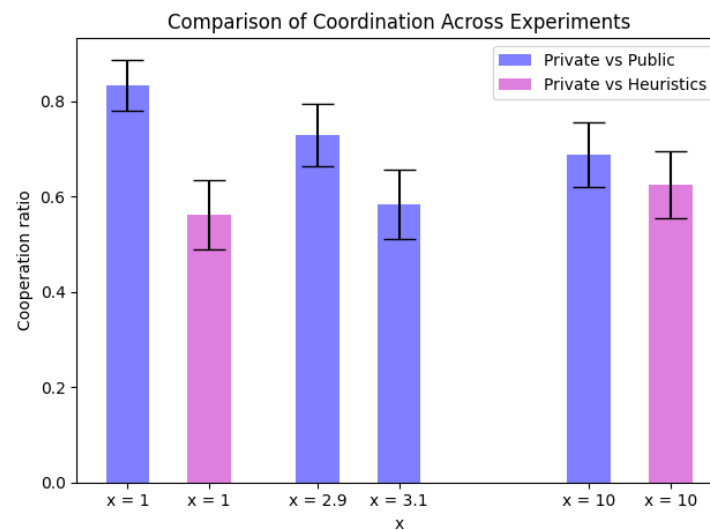


Figure 9. Coordination ratio depending on the value x in Parameterized game $|J| = 15$ iterations averaged over $|I| = 10$ rounds.

4.5. Sensitivity Analysis

To demonstrate the reliability of our results, we conducted a sensitivity analysis. This analysis focused on the parameters of the normal form game, as presented in the parametrized game, where the rewards matrix, shown in Table 3, gradually shifted between competitive and cooperative games. Moreover, using different normal-form games with known characteristics, we tested the LLM agent's adaptability through the different equilibria presented in Table 2. We also examined aspects of the entire system, including agent types and the environment, by varying the underlying LLM (GPT-3.5-turbo and GPT-4) and the number of game steps.

The performance of the underlying LLM and its effect on the private agent is presented in Figure 7, showing the clear advantage of the more advanced models when using techniques such as ICL and CoT. The number of game steps could also be considered part of the sensitivity analysis over discrete parameters, as the number of steps was unknown to the gameplaying agents beforehand. We exogenously stopped the game after a predetermined number of steps, and the agents did not memorize game-playing episodes, so there was no spillover effect between multiple runs. Once the message history buffer was complete, the round number had no strategic effects on the agents' behaviors, except random behavioral occurrences, which we can link to the stochastic nature of LLMs.

Due to our experimental setup, we did not have other parameters available to change. For example, the context length was fixed in the underlying LLM. The number of remembered historical iterations was maximized within the context window, so it was dependent on OpenAI's fixed parameters.

4.6. Limitations and Constraints

Ensuring consistent and explainable outputs from intelligent agents is crucial, because humans are fundamentally limited in understanding AI (artificial intelligence) behavior. Explainability is an essential aspect of AI safety, which we define as the ability of an AI system to stay within the boundaries of desired states, i.e., worst-case guarantees [54].

In non-adversarial scenarios, this issue is less concerning. However, the lack of explainability becomes a significant issue with adversarial agents capable of deceiving their opponents (e.g., humans or other agents) and exploiting them for their gain. In such cases, we must rely on AI explainability for safety [55].

In the context of LLM models, they deliver exceptional performance, due to their immense scale, with billions of parameters. However, their size poses a significant challenge to existing explainability methods. To ensure safety and explainability, constraints may

need to be imposed on the training and functioning of LLMs. These constraints can be integrated directly into the automated optimization (learning) process or applied indirectly through a human-in-the-loop approach [56].

Agents compress information received from the complex environment to store it in finite memory (context). The loss of information during this process leads to various phenomena recognized in information theory, such as echo chambers, self-deception, and deception symbiosis [57]. Moreover, since we studied the effect of deception as an emerging ability of LLM agents without formal information-theoretic models, developing formal models of deception, such as the Borden–Kopp model that relies on degradation, corruption, denial, and subversion, would be an interesting direction for future research [58].

5. Discussion

In this paper, we investigated the capabilities of large language model (LLM) agents in participating in a two-player repeated game. Furthermore, we introduce an augmentation to an LLM agent, referred to as the private agent, enabling it to engage in private contemplation (i.e., thoughts) regarding past and future interactions and to reason about future actions. Moreover, the private deliberation was concealed from its opponent in repeated games.

We utilized the partially observable stochastic game (POSG) framework to define the gameplay and formalized in-context learning (ICL) and chain-of-thought (CoT) prompting. In experiments, we examined the distribution of action choices based on conversation history. The results demonstrated that the private agent consistently identified a more favorable action, leading to a higher long-term payoff. When identifying their opponent type, both public and private agents performed subpar. LLM (GPT-4) encountered difficulties in generating random numbers from various diverse distributions when investigating the ability to sample from distributions. This suggests its limitations in effectively sampling from prior distributions and utilizing, for instance, a Bayesian estimator for action selection. Improving on the weakness of LLM agents in sampling from different probability distributions and finding (near) optimal action selection distributions in gameplay shows potential for future research.

Conducting simulations across various game settings, from competitive scenarios (e.g., prisoner's dilemma) to purely cooperative ones (e.g., Head-tail game), we found that augmenting an agent with the ability to privately deliberate on actions resulted in a superior overall performance and a clear advantage in competitive scenarios. Compared to the baseline agent, the private agent consistently outperformed or, at worst, matched its performance. In a direct comparison with the heuristic agent, which employed a tit-for-tat strategy in the prisoner's dilemma game, the private agent's performance was marginally lower. However, the heuristic agent's inability to communicate or deceive its opponent allowed the private agent to quickly discern its strategy, giving it a competitive edge. Moreover, the private agent's ability to deceive its opponent was noteworthy, securing a better overall score.

As part of our sensitivity analysis, we tested a gradual shift from a competitive to cooperative nature of the normal-form game, defined through a parameterized payoff matrix. The results suggested a high level of adaptability, except when close to the breaking point between the two dominant strategies. Additionally, we varied certain aspects of our system, including the underlying LLM, agent type, and the number of game steps. The more advanced LLM demonstrated greater differentiation of the proposed private agent than its counterparts, as it was implemented using ICL and CoT, which required a more capable model. Once the message history buffer was full, increasing the number of game steps did not yield any significant advantages in our case. However, if the context length limit baked into the underlying LLM was higher, this might produce different outcomes, which we leave open for future research.

For future work, our plan involves enhancing the private agent through additional fine-tuning. With this approach, we could further structure private thoughts and public

output, and align them with policy, such as to facilitate a more direct deception mechanism. Moreover, enhancing LLM agents with tools that allow, e.g., sampling from a probability distribution, Bayesian estimator calculation, and algorithm selection would greatly enhance strategies in multi-player games. Additionally, given the LLM agent's private deliberation results in gaming scenarios, we plan to explore its potential applications outside of gaming, including interactive simulations and decision support systems.

6. Conclusions

In conclusion, this research explored the potential of large language model (LLM) agents, specifically GPT-4, in two-player repeated games and introduced a novel augmentation: the private agent. This augmentation implemented through in-context learning (ICL) and chain-of-thought (CoT) allowed concealed private contemplation about past and future interactions, enhancing the agent's decision-making process. Utilizing the partially observable stochastic game (POSG) framework, ICL, and CoT prompting, our experiments revealed that the private agent consistently achieved higher long-term payoffs and outperformed the public (baseline) and heuristic agents in various game scenarios. However, the public and private agents struggled with identifying opponent types and sampling from diverse probability distributions, highlighting areas for future improvement.

The private agent's superior performance in competitive settings and ability to deceive opponents highlight its strategic advantages. Future research will focus on fine-tuning the private agent to enhance its deceptive capabilities and on exploring its applications beyond gaming, such as interactive simulations and decision support systems.

7. Limitations

While we showed that augmenting the LLM agent with private deliberation produced superior results overall in repeated games, there were still some limitations. Increasing the number of recall iterations in an LLM agent aids decision-making by providing a more extensive record of interactions with other agents [59]. However, when we increased the number of recall iterations, we concatenated the agent's generated output during each iteration, the length of which is depicted in Figure 6. Furthermore, with increased recall iterations, the context length became too large for GPT-4-0613, leading the model to either miss crucial information or engage in hallucination [60], negatively impacting its reasoning abilities. To address this issue, methods for computationally efficient extension of the context window, as proposed in [61,62], may need to be implemented.

8. Ethics Statement

This study entails the discussion and analysis of a simulated game setting, with any references to crime, animal torture, gender discrimination, or related actions strictly confined within the context of this game. The authors do not endorse violence or illegal activities in real-life scenarios. The game presented in this paper is designed for entertainment and research purposes, aiming to understand game mechanics, player behavior, and artificial intelligence. Moreover, it is important to emphasize that this study strictly adhered to all relevant ethical guidelines, maintaining the highest standards of research integrity.

Author Contributions: Software K.P., Writing—original draft preparation K.P., Investigation K.P., methodology M.B., resources M.B., writing—review and editing M.B., writing—review and editing M.B.B., writing—review and editing M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
POSG	Partially Observable Stochastic Game
OOP	Objective Oriented Programming
SLU	Spoken Language Understanding
CoT	Chain of Thought
ICL	In-Context Learning
PD	Prisoner's Dilemma
SH	Stag Hunt
GPT	Generative Pre-trained Transformer
AI	Artificial Intelligence

References

1. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
2. Høglund, S.; Khedri, J. Comparison Between RLHF and RLAI in Fine-Tuning a Large Language Model. Available online: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-331926> (accessed on 1 May 2024).
3. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
4. Creswell, A.; Shanahan, M.; Higgins, I. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv* **2022**, arXiv:2205.09712.
5. Meta Fundamental AI Research Diplomacy Team (FAIR); Bakhtin, A.; Brown, N.; Dinan, E.; Farina, G.; Flaherty, C.; Fried, D.; Goff, A.; Gray, J.; Hu, H.; et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science* **2022**, *378*, 1067–1074. [[PubMed](#)]
6. OpenAI. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
7. Park, J.S.; O'Brien, J.C.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. *arXiv* **2023**, arXiv:2304.03442.
8. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent abilities of large language models. *arXiv* **2022**, arXiv:2206.07682.
9. Andreas, J. Language models as agent models. *arXiv* **2022**, arXiv:2212.01681.
10. Li, G.; Hammoud, H.A.A.K.; Itani, H.; Khizbullin, D.; Ghanem, B. Camel: Communicative agents for “mind” exploration of large scale language model society. *arXiv* **2023**, arXiv:2303.17760.
11. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [[CrossRef](#)]
12. Poje, K.; Brcic, M.; Kovač, M.; Krleža, D. Challenges in collective intelligence: A survey. In Proceedings of the 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 22–26 May 2023; pp. 1033–1038.
13. Başar, T.; Olsder, G.J. *Dynamic Noncooperative Game Theory*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1998.
14. Isufi, S.; Poje, K.; Vukobratovic, I.; Brcic, M. Prismatic view of ethics. *Philosophies* **2022**, *7*, 134. [[CrossRef](#)]
15. Shoham, Y.; Leyton-Brown, K. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*; Cambridge University Press: Cambridge, UK, 2008.
16. Chawla, K.; Ramirez, J.; Clever, R.; Lucas, G.; May, J.; Gratch, J. Casino: A corpus of campsite negotiation dialogues for automatic negotiation systems. *arXiv* **2021**, arXiv:2103.15721.
17. Webb, T.; Holyoak, K.J.; Lu, H. Emergent analogical reasoning in large language models. *Nat. Hum. Behav.* **2023**, *7*, 1526–1541. [[CrossRef](#)] [[PubMed](#)]
18. Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; Sui, Z. A survey for in-context learning. *arXiv* **2022**, arXiv:2301.00234.
19. Fu, Y.; Peng, H.; Khot, T.; Lapata, M. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv* **2023**, arXiv:2305.10142.
20. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A survey of large language models. *arXiv* **2023**, arXiv:2303.18223.
21. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. Toolllm: Facilitating large language models to master 16,000+ real-world apis. *arXiv* **2023**, arXiv:2307.16789.
22. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.R.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. In Proceedings of the Thirty-Seventh Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Volume 36.

23. Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv* **2022**, arXiv:2207.05608.
24. Diji, Y.; Kezhen, C.; Jimeng, R.; Xiaoyuan, G.; Yawen, Z.; Jie, Y.; Yi, Z. Tackling vision language tasks through learning inner monologues. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 19350–19358.
25. Junkai, Z.; Liang, P.; Huawei, S.; Xueqi, C. Think Before You Speak: Cultivating Communication Skills of Large Language Models via Inner Monologue. *arXiv* **2023**, arXiv:2311.07445.
26. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the opportunities and risks of foundation models. *arXiv* **2021**, arXiv:2108.07258.
27. Kurvinen, E.; Koskinen, I.; Battarbee, K. Prototyping social interaction. *Des. Issues* **2008**, *24*, 46–57. [[CrossRef](#)]
28. Schön, D.A. *The Reflective Practitioner: How Professionals Think in Action*; Routledge: London, UK, 2017.
29. Gordon, M.L.; Zhou, K.; Patel, K.; Hashimoto, T.; Bernstein, M.S. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 8–13 May 2021; pp. 1–14.
30. Gordon, M.L.; Lam, M.S.; Park, J.S.; Patel, K.; Hancock, J.; Hashimoto, T.; Bernstein, M.S. Jury learning: Integrating dissenting voices into machine learning models. In Proceedings of the CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022; pp. 1–19.
31. Lee, M.; Srivastava, M.; Hardy, A.; Thickstun, J.; Durmus, E.; Paranjape, A.; Gerard-Ursin, I.; Li, X.L.; Ladhak, F.; Rong, F.; et al. Evaluating human-language model interaction. *arXiv* **2022**, arXiv:2212.09746.
32. Albrecht, S.V.; Christianos, F.; Schäfer, L. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*; The MIT Press: Cambridge, MA, USA, 2024.
33. Brookins, P.; DeBacker, J.M. Playing Games with GPT: What Can We Learn about a Large Language Model from Canonical Strategic Games? 2023. Available online: <https://ssrn.com/abstract=4493398> (accessed on 1 May 2024).
34. Guo, F. Gpt in game theory experiments. *arXiv* **2023**, arXiv:2305.05516.
35. Zhou, Z.; Liu, G.; Tang, Y. Multi-agent reinforcement learning: Methods, applications, visionary prospects, and challenges. *arXiv* **2023**, arXiv:2305.10091.
36. Zhang, K.; Yang, Z.; Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handb. Reinf. Learn. Control.* **2021**, *325*, 321–384.
37. Chen, Z.; Zhou, D.; Gu, Q. Almost optimal algorithms for two-player zero-sum linear mixture markov games. In Proceedings of the International Conference on Algorithmic Learning Theory, Paris, France, 29 March–1 April 2022; pp. 227–261.
38. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of hallucination in natural language generation. *ACM Comput. Surv.* **2023**, *55*, 1–38. [[CrossRef](#)]
39. Zhu, Z.; Cheng, X.; Li, Y.; Li, H.; Zou, Y. Aligner²: Enhancing joint multiple intent detection and slot filling via adjustive and forced cross-task alignment. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 19777–19785. [[CrossRef](#)]
40. Liu, B.; Lane, I. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv* **2016**, arXiv:1609.01454.
41. Aggarwal, M.; Hanmandlu, M. On modeling ambiguity through entropy. *Int. Trans. Oper. Res.* **2023**, *30*, 1407–1426. [[CrossRef](#)]
42. Jiang, H. A latent space theory for emergent abilities in large language models. *arXiv* **2023**, arXiv:2304.09960.
43. Liu, Q. Does gpt-4 play dice? *Chinaxiv* **2023**. [[CrossRef](#)]
44. Bravetti, A.; Padilla, P. An optimal strategy to solve the prisoner’s dilemma. *Sci. Rep.* **2018**, *8*, 1948. [[CrossRef](#)]
45. Tulli, S.; Correia, F.; Mascarenhas, S.; Gomes, S.; Melo, F.S.; Paiva, A. Effects of agents’ transparency on teamwork. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 22–37.
46. Harrison Chase. Langchain. Available online: <https://github.com/langchain-ai/langchain> (accessed on 5 April 2024).
47. Fudenberg, D.; Levine, D.K. *The Theory of Learning in Games*; MIT Press: Cambridge, MA, USA, 1998; Volume 2.
48. Neyman, A. Correlated equilibrium and potential games. *Int. J. Game Theory* **1997**, *26*, 223–227. [[CrossRef](#)]
49. Daskalakis, C.; Goldberg, P.W.; Papadimitriou, C.H. The complexity of computing a nash equilibrium. *Commun. ACM* **2009**, *52*, 89–97. [[CrossRef](#)]
50. Iancu, D.A.; Trichakis, N. Pareto efficiency in robust optimization. *Manag. Sci.* **2014**, *60*, 130–147. [[CrossRef](#)]
51. van der Rijt, J.-W. The quest for a rational explanation: An overview of the development of focal point theory. In *Focal Points in Negotiation*; Springer International Publishing: New York, NY, USA, 2019; pp. 15–44.
52. Thawani, A.; Pujara, J.; Ilievski, F. Numeracy enhances the literacy of language models. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Virtual Event, 7–11 November 2021; pp. 6960–6967.
53. Spithourakis, G.P.; Riedel, S. Numeracy for language models: Evaluating and improving their ability to predict numbers. *arXiv* **2018**, arXiv:1805.08154.
54. Došilović, F.K.; Brcic, M.; Hlupić, N. Explainable artificial intelligence: A survey. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 0210–0215.
55. Brcic, M.; Yampolskiy, R. V. Impossibility Results in AI: A survey. *ACM Comput. Surv.* **2023**, *56*, 1–24. [[CrossRef](#)]

56. Longo, L.; Brcic, M.; Cabitza, F.; Choi, J.; Confalonieri, R.; Del Ser, J.; Guidotti, R.; Hayashi, Y.; Herrera, F.; Holzinger, A.; et al. Explainable artificial intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Inf. Fusion* **2024**, *2024*, 102301. [[CrossRef](#)]
57. Enßlin, T.; Kainz, V.; Boehm, C. A Reputation Game Simulation: Emergent Social Phenomena from Information Theory. *Ann. Der Phys.* **2022**, *534*, 2100277. [[CrossRef](#)]
58. Carlo, K.; Kevin, B.K.; Bruce, I.M. Information-theoretic models of deception: Modelling cooperation and diffusion in populations exposed to “fake news”. *PLoS ONE* **2018**, *13*, e0207383.
59. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
60. Azamfirei, R.; Kudchadkar, S.R.; Fackler, J. Large language models and the perils of their hallucinations. *Crit. Care* **2023**, *27*, 120. [[CrossRef](#)] [[PubMed](#)]
61. Peng, B.; Quesnelle, J.; Fan, H.; Shippole, E. Yarn: Efficient context window extension of large language models. *arXiv* **2023**, arXiv:2309.00071.
62. Li, R.; Xu, J.; Cao, Z.; Zheng, H.T.; Kim, H.G. Extending Context Window in Large Language Models with Segmented Base Adjustment for Rotary Position Embeddings. *Appl. Sci.* **2024**, *14*, 3076. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.