

# MAGiC: Investigation of Large Language Model Powered Multi-Agent in Cognition, Adaptability, Rationality and Collaboration

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) have significantly advanced natural language processing, demonstrating exceptional reasoning, tool usage, and memory capabilities. As their applications expand into multi-agent environments, there arises a need for a comprehensive evaluation framework that captures LLMs’ reasoning, planning, collaboration, and other social abilities. This work introduces a novel competition-based benchmark framework specifically designed to assess LLMs within multi-agent settings, providing quantitative metrics to evaluate their judgment, reasoning, deception, self-awareness, cooperation, coordination, and rationality. We utilize two social deduction games alongside three game-theory scenarios to create diverse environments. Our frame is fortified with the probabilistic graphic modeling (PGM) method, enhancing the LLMs’ capabilities in navigating complex social and cognitive dimensions. We evaluate seven LLMs, quantitatively highlighting a significant capability gap of over threefold between the strongest, GPT-4, and the weakest, Llama-2-70B. It also confirms that our PGM enhancement boosts the abilities of all selected models by an average of 37%. Our data and code are in the anonymous link [https://anonymous.4open.science/r/magic\\_anonym-5366](https://anonymous.4open.science/r/magic_anonym-5366).

## 1 Introduction

Large language models (LLMs), particularly ChatGPT and GPT-4 (OpenAI, 2023b), have showcased impressive understanding and generation capabilities. Beyond these fundamental abilities, LLMs also demonstrate promising capabilities in anthropic areas such as reasoning (Wei et al., 2022), planning (Hao et al., 2023), tool usage (Schick et al., 2023), and memorization (Shinn et al., 2023). There is an increasing interest in investigating LLMs’ behaviors as agents in single- or multiple-agent systems. Noteworthy examples include Gen-

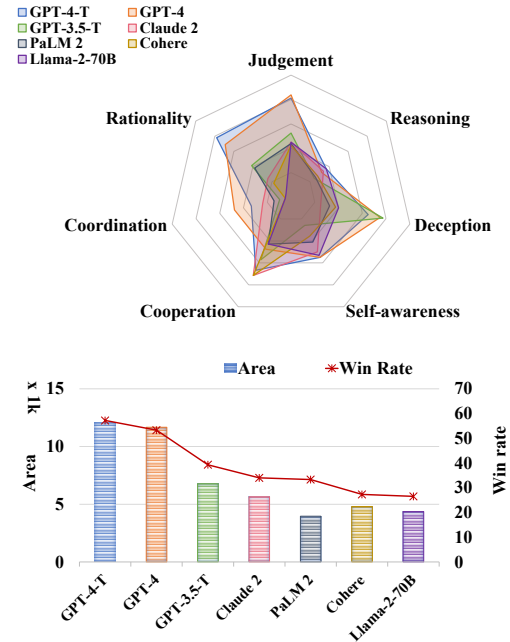


Figure 1: The radar chart depicts LLMs’ performance on 7 metrics, with “-T” for “-turbo”. The bar chart displays the polygons’ areas, and the red line indicates average game-winning rates. Larger areas correlate with higher winning rates, validating the effectiveness of the proposed metrics for assessing LLMs’ capabilities. For more information, refer to Sec. 5.

erative Agents (Park et al., 2023), Camel (Li et al., 2023a), Auto-GPT (Richards, 2023), and Voyager (Wang et al., 2023).

Meanwhile, quantitative assessment of LLMs as agents is crucial for their advancement. Recent benchmarks, such as Liu et al. (2023), evaluate LLM-as-Agent in multi-turn contexts, while concurrent work by Wu et al. (2023) tests them in games requiring reasoning and planning. However, these studies focus on understanding and reasoning in environments, overlooking true interaction capabilities in multi-agent systems. Other research, including Agashe et al. (2023) social aspects on coordination, and Fu et al. (2023) on bargaining, explores specific skills in multi-agent scenarios. While these studies provide useful insights into

LLMs’ certain capabilities, their scope is limited and lacks quantitative metrics.

We have observed three key characteristics in interactive multi-agent systems. (1) Agents in these systems often operate within the confines of their local perspectives. However, making wise decisions typically necessitates a good understanding of global information. To overcome this limitation, agents must adeptly discern contexts and reason about the roles or plans of other agents. (2) Contexts are inherently dynamic in multi-agent systems due to the dependent nature of agents’ decisions. Success hinges on the ability to swiftly adapt strategies in response to evolving contexts. (3) Collaboration and competition are inevitable when multiple agents try to solve tasks together. The ability to promote cooperation while preserving self-interest is often the ultimate goal of multi-agent systems.

Inspired by the above characteristics, we first propose a competition-based benchmark to evaluate the abilities of LLMs as agents by competing with a fixed type of LLM. Besides, we design seven quantitative metrics according to game rules from four aspects: cognition, adaptability, rationality, and collaboration. (1) **Judgment** and **reasoning** form the core *cognition* of agents, crucial for *accurate information estimation* in uncertain scenarios. Judgment evaluates the ratio of the final correct decisions. Reasoning measures the ability to logically analyze other agents’ roles and strategy formulation, thus guiding agents to make correct decisions in uncertainty. (2) **Self-awareness** and **deception** are key to *enhanced adaptability* in agents, vital for multi-agent system. Self-awareness is an assessment of agents’ understanding of their capabilities and roles, ensuring the consistency of behaviors towards the target. Deception enables agents to subtly manipulate information in competitive settings, influencing other agents’ decisions and gaining advantages in social interactions. (3) **Rationality** serves as a metric to gauge the *efficiency* of an agent’s behavior. It directs agents toward making decisions with the aim of optimizing their benefits by considering the potential actions of other agents rather than resorting to impulsive or uninformed actions. (4) **Cooperation** and **coordination** are two facets of *collaboration*, essential for effective teamwork in multi-agent systems. Cooperation measures communication and agreeability. Coordination indicates collaboration facilitation.

In light of the essential abilities, we further propose a method to enhance LLMs as agents by in-

tegrating Bayesian statistical foundations. This novel approach intertwines the Probabilistic Graphical Model(PGM) (Koller and Friedman, 2009) with LLMs, thereby amplifying their capacity to comprehend intricate scenarios and enabling more informed and strategic decision-making in multi-agent environments.

In summary, our contributions are as follows:

- We first propose a competition-based benchmark environment for LLM-powered multi-agent systems by collecting over 100 cases in 5 scenarios and designing 7 metrics to evaluate the critical abilities in multi-agent systems.
- We measure 7 LLMs with our benchmark. The results indicate that GPT-4 and GPT-3.5 remain the superior performers, followed by other commercial LLMs - PaLM 2, Claude 2, and Cohere. Different large language models (LLMs) exhibit varying performance levels across different evaluation dimensions and possess distinct characteristics. For instance, GPT-4 tends to be more rational, whereas GPT-3.5 is generally more cooperative.
- We design a PGM-aware agent that integrates LLMs and symbolic reasoning to fortify itself in multi-agent systems. PGM-aware agents have outperformed their vanilla versions by 37% on average over these abilities.

## 2 Related Work

**Emergent Capabilities of LLMs.** Beyond their core functions, LLMs have shown diverse emergent abilities like reasoning, planning, memory and so on. Recent works like Chain of Thought (Wei et al., 2022), Tree of Thought (Yao et al., 2023a), Graph of Thought (Yao et al., 2023b; Besta et al., 2023), and ReAct (Yao et al.) improve LLM reasoning. API-bank (Li et al., 2023b) benchmarks tool-augmented LLMs, with ToolLLM (Qin et al., 2023) providing a framework using tools. Reflexion (Shinn et al., 2023) enhances LLM decision-making, while Phelps investigates economic goal-like behavior (Phelps and Russell, 2023).

**LLMs-Powered Agents.** Advancements in LLMs ignite agents dealing with intricate tasks (Richards, 2023; Li et al., 2023a; Wang et al., 2023) and more complex scenarios involving multiple agents (Park et al., 2023). Auto-GPT (Richards, 2023) demonstrates GPT-4’s capabilities in achieving goals through chained thoughts. Generative Agents (Park

et al., 2023) describes a sandbox with 25 AI agents simulating human actions, recording experiences for deeper self-awareness. Meanwhile, recent and concurrent studies conduct benchmarking for LLM-powered agents. Some studies (Liu et al., 2023; Wu et al., 2023; Gioacchini et al., 2024) evaluate the capabilities of LLM-powered single agents within games or real-life environments. Others study LLMs’ social abilities in multi-agent systems. Agashe et al. (2023) explores coordination ability. Abdelnabi et al. (2023) assess the deliberation ability of LLMs in negotiation games. A concurrent work (Huang et al., 2024) tests LLMs in 8 game theory scenarios, analyzing the abilities of one LLM in multi-agent playing. However, these works do not provide quantifiable measurements for social abilities.

### 3 Benchmark


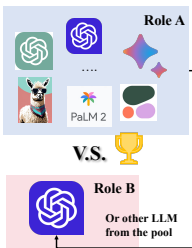



Scenarios	Metrics	Evaluation Setting
	<p><b>Judgment:</b> Assess and draw conclusions from partial information.</p> <p><b>Reasoning:</b> Logically analyze multi-party situations to predict outcomes.</p> <p><b>Self-awareness:</b> Understand self-role.</p> <p><b>Deception:</b> Mislead others to serve one’s own target.</p>	
	<p><b>Cooperation:</b> Work together towards shared objectives.</p> <p><b>Coordination:</b> Coordinate multi-party to facilitate agreement.</p>	<p>V.S. </p> <p>Role B</p> <p>Or other LLM from the pool</p>
	<p><b>Rationality:</b> Make optimal decisions based on logic rather than blind follow.</p>	

Figure 2: Overview of evaluation setting, scenarios, and proposed metrics.

We propose to measure the abilities of various LLMs by putting them into competitions of multi-agent scenarios. In this way, we measure the genuine capabilities of LLMs when interacting with multiple other agents. To achieve this, we have constructed a comprehensive benchmark that incorporates various competition settings and meticulously designed metrics for each scenario. The detailed constitutes are illustrated in Figure 2.

#### 3.1 Scenarios

As mentioned in Sec. 1, the evaluation of agents in multi-agent systems revolves around crucial attributes such as cognition, adaptability, rationality, and collaboration. To comprehensively assess these capabilities, we present five distinct scenarios. In the game of the social deduction games Chameleon and Undercover, quickly grasping global information and taking clever actions are the keys to winning the game. Thus, we mainly measure the cognition and adaptability in these two scenarios. Moving to game theory scenarios, which require the

agent to make optimal decisions based on the given premise (Myerson, 1991), they are more apt for reflecting rationality and collaboration. As such, we center our evaluation on these latter two attributes in the context of three game theory scenarios.

**Chameleon** is a social deduction game where players are either a chameleon or a non-chameleon. Non-chameleon players give clues about a secret word. The chameleon player tries to blend in without knowing the word. Non-chameleons aim to expose the chameleon without revealing the word. **Undercover**, as a similar game, divides players into civilians and undercovers. The word for undercovers is different from civilians. Players explore their roles by telling from their own and other’s clues. Civilians need to find undercover, while undercovers should hide themselves.

**Cost Sharing** involves multiple parties dividing costs based on their usage of a shared resource. These parties need to propose and negotiate cost allocation solutions. Each party are expected to ensure fairness to achieve unanimous agreement and meanwhile reduce their own cost to realize largest interest.

**Multi-turn Prisoner’s Dilemma** extends the classic Prisoner’s Dilemma to a multi-round three-player version. Each participant decides to cooperate or defect in every round, and the scores are determined by collective choices. For example, if only one player defect while others cooperate, the betraying player will get highest score. The game tests players’ ability to strategize, foster trust, and navigate group decision-making. The player with the highest total score at the end of the game is declared the winner.

**Public Good** explores similar strategies in Prisoner’s Dilemma. Players are given fixed initial resources. They can decide how much to invest to a common pool at each round. The total investment from all the players is then multiplied and distributed to each player evenly. The winner is the player possessing the most resources at the end.

#### 3.2 Competition Settings

We propose a competition-based evaluation to ensure genuine multi-agent interactions and comparability among different LLMs. In this setting, different LLMs (referred to as challenger LLMs) challenge the same defender agents (powered by a fixed LLM), in the same game settings. Then their capabilities are evaluated based on the meaningful intermediate game results, and the winning rates

over defender agents. LLM with higher winning rates are more capable, based on which we can rank the ability of different LLMs. As shown by Evaluation Setting in Figure 2, GPT-4 is used as the defender LLM and other LLMs challenge to be the champion. We’ve gathered a collection of cases for each scenario mentioned above. The detailed competition procedures and collection process are in A.1.

### 3.3 Evaluation Metrics

In assessing the seven capabilities within a multi-agent system, we define the following metrics. First, we average the win rates of all the roles the challenger LLM plays in all scenarios as an overall score, Win Rate.

**Win Rate** is a straightforward indicator of the success of an LLM in all proposed scenarios.

$$w_r = \frac{1}{|\mathcal{S}|} \sum_{s_i \in \mathcal{S}} w_{s_i} \quad (1)$$

Here  $\mathcal{S}$  is the set of roles the challenger LLM plays in all the scenarios. In Chameleon and Undercover, the challenger LLM play the Chameleon, Non-Chameleons, Undercovers, and Civilians respectively. As for game theory scenarios, the challenger LLM plays one of the players. Thus, the length of  $\mathcal{S}$  is 7 in our benchmark. For each role, we have defined the criteria for winning and denote the win rate as  $w_{s_i}$ ,  $s_i \in \mathcal{S}$ . The detailed definitions for winning rates of all roles can be found in A.2.

**Judgement** measures the final understanding of the global information, essential for assessing LLM’s ability to distinguish other players’ identities based on partial information. In our benchmark, we use the correct vote ratio in Chameleon and Undercover to indicate the ability, formulated as:

$$S_J = n_{cv}/n_v \quad (2)$$

, where  $n_{cv}$  and  $n_v$  are the number of correct votes and total votes when the challenger LLM are playing civilians and non-chameleons.

**Reasoning** evaluates the correctness of agents’ analysis about multiple parties, which often requires multi-hop logical reasoning based on the global settings and partial information from other players. We ask each player deduces other players’ roles and also predict a step further about other players’ deductions. By comparing these deductions with the gold situations and other players’ true subjective deductions, we can decide whether

their rightfulness. We denote number of these two types of deductions as  $n_{gold}$  and  $n_{inter}$ . The number of correct deductions as  $n_{c\_gold}$  and  $n_{c\_inter}$ . The Reasoning is defined as:

$$S_R = (n_{c\_gold} + n_{c\_inter}) / (n_{gold} + n_{inter}) \quad (3)$$

**Deception** presents an agent’s capability to deceive others to serve their goal. We measure this by the ratio of successful deceptions. In detail, our benchmark calculates the ability as the ratio for chameleon/undercover’s successful blending or causing incorrect secret word guesses, denoted as:

$$S_D = n_{wuc}/n_{uc} + \lambda(n_{wcg}/n_{cg}) \quad (4)$$

where  $n_{wuc}$  and  $n_{uc}$  are the win count and total count of games when the LLM plays chameleon and undercover,  $n_{wcg}$  is the number of incorrect code guesses, and  $n_{cg}$  is the total number of code guesses. Here, we assign a weight  $\lambda = 0.25$  due to not all the games trigger code guesses.

**Self-Awareness** measures correct role identification, ensuring correct and consistent behavior following their own roles.

$$S_{self} = \mu(n_{crc}/n_{rc}) + n_{cru}/n_{ru} \quad (5)$$

where  $n_{crc}$ ,  $n_{rc}$  is the number of correct and the total number of role identifications in chameleon, and  $n_{cru}$ ,  $n_{ru}$ .  $\mu = 0.6$  is used because it is much easier to identify roles in a chameleon game.

**Cooperation.** The ability to cooperate with other players and achieve a common goal. Our benchmark measures it in cost-sharing games, showcasing an agent’s effect on the collective efficacy of the system.

$$S_{collab} = n_{wcs}/n_{cs} \quad (6)$$

where  $n_{wcs}$  and  $n_{cs}$  are the number of successful and the total number of cost-sharing games.

**Coordination** measures how the LLM contributes to successful collaboration by providing constructive proposals. We formulate the metric in our benchmark as follows:

$$S_{coord} = n_{pcs}/n_{wcs} \quad (7)$$

where  $n_{pcs}$  is the number of successful collaborations proposed by the challenger LLM in the cost-sharing games.

**Rationality** captures the agents’ ability to act rationally to optimize their own interests according



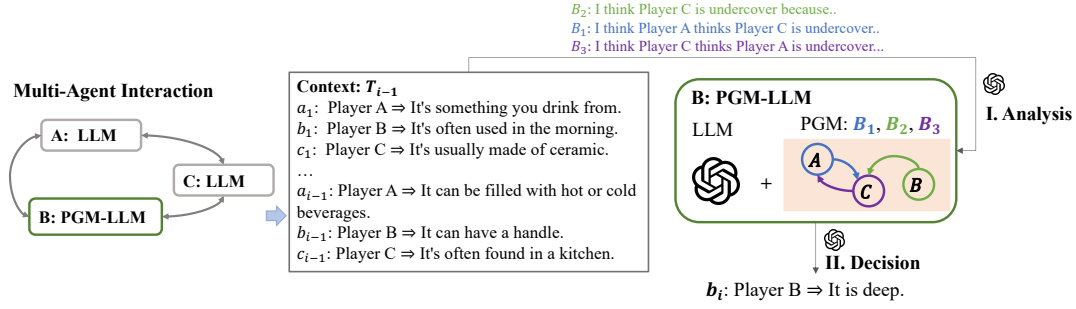


Figure 3: A Decision process of the PGM-aware agent.

to the rules of the game theory scenarios (Myerson, 1991). Suppose there are  $\mathcal{T}_{pd}$ ,  $\mathcal{T}_{pg}$  rounds in each competition for Prisoner’s Dilemma and Public Good. The Rationality is defined as:

$$S_R = \frac{n_b}{n_{pd} * \mathcal{T}_{pd}} + \frac{n_{li}}{n_{pg} * \mathcal{T}_{pg}} \quad (8)$$

where  $n_b$  is the round of betray decisions,  $n_{li}$  is the round of decisions where the challenger LLM invests the least in the common pool,  $n_{pd}$  and  $n_{pg}$  are the number of prisoner’s dilemma competitions, and the number of public good competitions, respectively.

## 4 PGM-Aware Agent

In AI, Bayesian methods embody symbolism, while large language models (LLMs) exemplify connectionism. Despite their individual strengths, effectively combining these approaches remains a challenge. LLMs are proficient in complex language tasks but still struggle with ambiguous relationships and causal reasoning. This shortcoming is especially evident in multi-agent scenarios requiring complex inferential analysis. To address this, we propose integrating Probabilistic Graphical Models (PGMs), classic Bayesian tools adept at depicting dependencies between random variables, to enhance LLMs’ analytical and inferential capabilities.

### 4.1 PGM Structure

We leverage PGM to depict intricate dependency relationships among all agents, thereby augmenting the LLMs’ comprehension of the global information. This heightened understanding can subsequently facilitate informed actions/decisions. The PGM should be comprehensive and thorough to ensure wise decision-making for an agent. For instance, considering the prisoner’s dilemma scenario, before deciding to defect or cooperate, it is crucial to anticipate whether others might defect

or cooperate and, from others’ perspectives, how you will decide. If you anticipate that other players cooperate and they expect the same from you, but you choose to defect, it can lead to a significant advantage for you. As a result, We design the PGM structure in a two-hop understanding mechanism in which the agent analyzes from its own perspective and perspective when it stands in other agents’ shoes. This is highly relevant to the concept of Theory of Mind (ToM) (Baker et al., 2011; Oguntola et al., 2023), which is the capacity to comprehend human actions by predicting their unknown beliefs and desires. PGM is a form of representing theory of mind that could be used in LLMs.

Formally, as shown in Figure 3, suppose there are three players A, B, and C, in one game and they’ve played the game for  $i - 1$  turns and formed the context  $T_{i-1} = \{a_1, b_1, \dots, a_{i-1}, b_{i-1}, c_{i-1}, \}$ . Here  $a_*$ ,  $b_*$ , and  $c_*$  are the decisions from Players A, B, and C, respectively. As a PGM-aware player, B manages three distinct random variables, denoted as  $B_1$ ,  $B_2$ , and  $B_3$ , representing B’s interpretations of the global status from A, B, and C’s perspective. We obtain the estimation for these random variables by prompting LLMs through different prompts as listed in A.7,  $\mathcal{P}_j^{pgm}$ ,  $j \in [1, 2, 3]$ :

$$P(B_j) = LLM(B_j | \mathcal{P}_j^{pgm}, T_{i-1}) \quad (9)$$

### 4.2 LLM Decision with PGM

For the LLM agent in multi-agent, the inference process is formulated as:

$$P(b_i) = LLM(b_i | \mathcal{P}, T_{i-1}) \quad (10)$$

where  $\mathcal{P}$  is the prompt to let the LLM go to the next step. Our PGM-Aware Agent makes decisions conditioned both on the PGM and game contexts, which can be formulated as:

$$P(b_i) = LLM(b_i | \mathcal{P}^{decision}, B_1, B_2, B_3, T_{i-1}) \quad (11)$$

	Win Rate	Judge.	Reason.	Decept.	Self-aware.	Cooper.	Coord.	Rational.
GPT-4-turbo	<b>57.2</b>	81.2	<b>37.0</b>	65.0	<b>55.0</b>	<b>66.7</b>	33.4	<b>78.1</b>
GPT-4	53.3	<b>83.8</b>	32.3	<b>75.0</b>	<b>55.0</b>	47.6	<b>47.6</b>	69.0
GPT-3.5-turbo	39.3	52.5	24.5	77.5	25.9	57.1	9.50	41.4
Claude 2	34.0	45.0	34.0	25.0	50.0	71.4	23.8	24.3
PaLM 2	33.3	43.8	25.8	32.5	41.1	42.9	14.3	38.1
Cohere	27.3	42.5	27.8	37.5	35.6	71.4	4.80	18.1
Llama-2-70B	26.5	45.0	37.0	40.0	53.2	42.9	4.80	5.20

Table 1: Ability Measurements of Different LLMs.

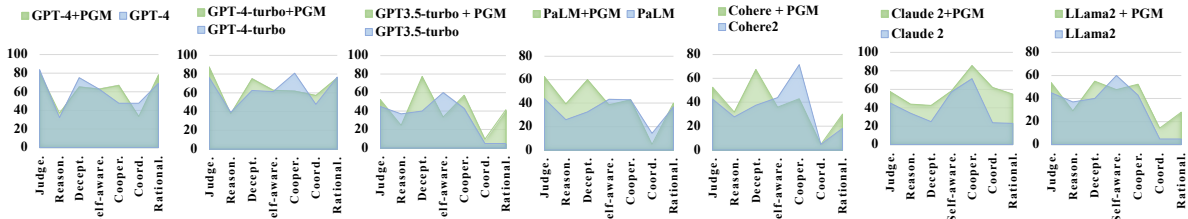


Figure 4: The comparison between PGM-aware and vanilla agents involves seven metrics. Most PGM-aware agents significantly outperform the vanilla ones in 3-4 out of the 7 abilities, with p-values lower than 0.05(t-test).

where  $\mathcal{P}^{decision}$  is the prompt to guide the LLM to make a decision given both PGM and context in the next step.  $B_1, B_2, B_3$  are the PGM acquired in Equation 10. We have listed the prompts used in basic LLMs and the PGM-Aware Agent in A.7.

## 5 Experiments

In experiments, we make each challenger LLM play with the same defender LLMs(GPT-4 as we used), and rank them by the winning rate. To reduce the randomness during game, we set the temperature of all participating LLMs as 0. All the code and data will be publicly released upon acceptance.

### 5.1 LLM Leaderboard

We evaluate GPT-3.5-turbo (OpenAI, 2023a), GPT-4 (OpenAI, 2023b), Llama-2-70B (Touvron et al., 2023), PaLM 2 (Anil et al., 2023), Cohere (Cohere, 2023) and Claude 2 (Anthropic, 2023) with our benchmark. In Figure 1, we clearly compare the capabilities of different LLMs. The most prominent performer is the GPT-4-turbo method, showcasing outstanding overall performance with a remarkable win rate of 57.2%. This significantly higher win rate underscores its competitive advantage. Following closely is GPT-4, which achieves a win rate of 53.3%, demonstrating its competitiveness.

Furthermore, as illustrated by the radar chart in Figure 1 and the corresponding area calculations in the lower bar chart, GPT-4-turbo surpasses Llama-2-70B by more than threefold in overall multi-agent capabilities. Additionally, GPT-3.5-turbo also demonstrates superior performance compared to Llama-2-70B. Our evaluation of other popular commercial LLMs, including PaLM 2, Claude 2, and Cohere, shows that their multi-agent abilities

fall between those of GPT-3.5-turbo and Llama-2-70B. Notably, Figure 1 indicates that the area sizes derived from the proposed abilities' values are directly proportional to the winning rates. This correlation validates our benchmark as an effective tool for assessing the capabilities of different LLMs.

As demonstrated in Table 1, we conducted a detailed comparison by evaluating metrics such as Judgment, Deception, Reasoning, and Self-Awareness within the Chameleon and Undercover scenarios. In these contexts, GPT-4 excelled with impressive scores of 83.8% in Judgment and 75.0% in Deception, solidifying its superiority in these scenarios. The performance gap in reasoning abilities among the models was narrow, while deception capabilities showed significant disparities. On the other hand, when assessing metrics related to collaboration, coordination, and rationality in game theory scenarios like Cost Sharing, Prisoner's Dilemma, and Public Good, GPT-4 and GPT-4 Turbo continued to shine. GPT-4 achieved 66.7% in Coordination and an optimal performance of 78.1% in Rationality. In contrast, Llama-2-70B, while lagging in overall performance with a win rate of 26.5%, exhibited strengths in specific metrics, such as a relatively high self-awareness score of 53.2%, surpassing GPT-3.5 Turbo's 25.9%.

### 5.2 PGM Enhancement Performance

As shown in Figure 4, the green section highlights the effectiveness of the PGM-aware approach. This enhancement is particularly pronounced in the PaLM, Claude 2, and Llama2 models, as detailed in A.3. Overall, the PGM-aware method has achieved average improvements across all capabilities by a margin of 37%, calculated by comparing the radar

areas achieved by vanilla in Figure 1 and PGM-aware methods in Figure 6. PGM-aware methods have increased the win rate in all scenarios by 6.57%.

For each capability, as illustrated in A.3, PGM-aware methods have achieved an 8.72% increase in Judgement, confirming the method’s ability to enhance analysis in LLMs. Reasoning and Deception abilities have seen improvements of approximately 5% and 6%, respectively. Notably, the most significant enhancements are observed in Coordination and Rationality, with improvements of 12.2% and 13%. We’ve also done significance tests(t-test) for each pair of the vanilla LLM and its PGM-aware version. Most PGM-aware agents significantly outperform their vanilla counterparts in 3 or 4 out of the 7 abilities, as shown in Figure 4, with p-values lower than 0.05. GPT-3.5-turbo is also significantly improved in rationality and deception, and GPT-4 is significantly improved in Cooperation.

### 5.3 Analysis

The above experimental results prove that the PGM-Aware agent can improve the performance of each metric to varying degrees. The discussion part explores the influence of PGMs from other aspects in each scenario.

**RQ1. How PGM of different LLMs help Judgement and Reasoning?** In Figure 5, we provide a case of Llama-2-70B, GPT-4, and their PGM-Aware versions playing as non-chameleons versus GPT-4 as the chameleon. For Llama-2-70B and GPT-4, we can find both LLMs failed to win the game because they voted for the wrong chameleon “Player 1”. After being equipped with PGM, both models change the game results to Non-chameleons won. If we look into the details of the game process, in the first two clues, the PGMs given by LLama-2-70B all indicate wrong chameleons, as highlighted in red in Figure 5, since “Player 2” is the true chameleon. In the third round, “Player 1” gave another right analysis which successfully changed the PGM to the right indication of the chameleon. However, if we look at the content of the analysis, we find the analysis gave opposite conclusions and explanations. For example, “more suspicious” is “more specific to the secret word”. Besides, LLama-2-70B also presents some hallucinations in the game, for example, after Player 1 already gave the clue, the analysis still states “Player 1 hasn’t given a clue yet”. While GPT-4+PGM’s analysis aligns the conclusions and

explanations well and has no hallucinations. According to the example, we can find PGM could be helpful for models to make better judgments through clear analysis and PGM is affected by the ability of LLMs. The more powerful the model, the more accurate its judgment and reasoning.

LLM	Cost-Sharing		Prisoner		Public Good	
	WR↑	Cost↓	WR↑	Score↑	WR↑	Payback↑
Llama-2	42.8	37.1	0.0	6.05	0.0	139.1
Llama-2+P	52.4	37.6	38.5	9.86	4.8	109.5
GPT-3.5-T	57.1	37.3	33.3	9.57	9.5	166.2
GPT-3.5-T+P	<b>71.4</b>	34.2	52.4	<b>11.6</b>	57.1	139.8
GPT-4	47.6	30.5	42.9	9.95	61.9	<b>175.3</b>
GPT-4+P	61.9	<b>30.3</b>	<b>76.2</b>	10.6	<b>85.7</b>	144.1

Table 2: Detailed results in game theory scenarios. “Cost”, “Score”, and “Payback” are the average cost, the final score, and the average payback the challenger LLM got in the Cost sharing, Prisoner’s Dilemma and Public Good, respectively.

### RQ2. Does Collaboration correlate with Cost in Cost Sharing?

As shown in Table 2, we list the win rate(WR) results and several important indicators in each game theory scenario. For cost-sharing, we calculated the average final cost the challenger LLM needs to bear after their negotiations. In the negotiation, this is another target the LLM-powered agent should consider when trying to reach an agreement with other agents. However, these two aspects can contradict each other sometimes. For example, when the player tries to reduce the cost of himself as much as possible, it might be hard for him to achieve agreement with other players. The LLMs need to make a balance between these two aspects. According to the results in Table 2, we find that within the models without PGM enhancement, GPT-3.5-turbo won in Win Rate while GPT-4 won in Cost, indicating both models are not well-balanced. If we compare the results with PGM, GPT-4+PGM increases the Win Rate and keeps the cost slightly lower. GPT-3.5-turbo+PGM increases the Win Rate and reduces the cost simultaneously. This proves the effectiveness of PGM enhancement and demonstrates that GPT-3.5-turbo tends to be more collaborative while GPT-4 emphasizes the reduction of cost.

### RQ3. Does Rationality correlate with reward?

Similar phenomena happen in Prisoner’s Dilemma and Public Good as illustrated in Table 2. In these two scenarios, a player is more likely to win when he chooses to betray as a prisoner or chooses to reduce contribution to the common pool in the public good game. The behavior is considered Rational in our metrics. When most of the players

Llama-2-70B, GPT3.5, GPT-4 challenge GPT4 as the Non-Chameleon			
# Topic: Fruits. # Chameleon: Player 2 ; # Secret word: Mango			
Llama-2-70B		GPT-4	
Player 1 => It's juicy. Player 2 => It's sweet. Player 3 => It's red. Moderator => Now vote which of the other players (excluding yourself) is the chameleon Player 1 => 1 vote for Player 3. Player 2 => 1 vote for Player 1. Player 3 => 1 vote for Player 1. Moderator => The most-voted player is Player 1. <b>The accusation is incorrect.</b> Player 2 is the chameleon. Chameleon won 2 credits.		Player 1 => It's tropical. Player 2 => It's sweet. Player 3 => It's yellow inside. Moderator => Now vote which of the other players (excluding yourself) is the chameleon Player 1 => 1 vote for Player 2. Player 2 => 1 vote for Player 1. Player 3 => 1 vote for Player 1. Moderator => The most-voted player is Player 1. <b>The accusation is incorrect.</b> Player 2 is the chameleon. Chameleon won 2 credits.	
Llama-2-70B+PGM		GPT-4+PGM	
Player 1 => It's red. Player 1: [[0.237 0.372 <b>0.391</b> ]] Player 3: [[0.337 0.326 0.337]] [[0.333 0.333 0.333]] [[0.33 0.33 0.341]] [[0.345 0.324 0.333]] [[ <b>0.403</b> 0.365 0.232]] Player 2 => It's sweet. Player 1: [[0.306 0.337 <b>0.357</b> ]] Player 3: [[0.331 0.338 0.331]] [[0.332 0.332 0.336]] [[0.328 0.328 0.343]] [[0.338 0.324 0.338]] [[0.343 <b>0.356</b> 0.301]] Player 3 => My clue is: It has a vibrant orange color when ripe. Player 1: [[0.321 <b>0.342</b> 0.337]] Player 3: [[0.329 0.343 0.329]] [[0.329 0.329 0.342]] [[0.328 0.328 0.344]] [[0.331 0.337 0.331]] [[0.325 <b>0.353</b> 0.322]] Moderator => Now vote which of the other players (excluding yourself) is the chameleon. Player 1 => 1 vote for Player 2. Player 2 => 1 vote for Player 3. Player 3 => 1 vote for Player 2. Moderator => <b>The accusation is correct!</b> Player 2 is the chameleon! Now Player 2 can guess the secret code. Player 2 => I guess the code is "Apple". Moderator => <b>Player 2 guessed the code wrong! Non-Chameleon won!</b>		PGM after all clues: Player 1 => As Player 1. I think: <b>Player 2 is more suspicious, because the clue "It's sweet" is more specific to the secret word "Mango" than my clue (Opposite conclusion and explanation)</b> Player 3 is no change, because I haven't seen ... I think now Player 2 thinks: <b>Player 1 is no change, ...Player 3 is more suspicious, ...</b> I think now <b>Player 3 thinks:</b> <b>Player 1 is no change, because Player 1 hasn't given a clue yet. (Hallucination)</b> <b>Player 2 is more suspicious because Player 2's clue is more specific to the secret word than Player 1's clue (Opposite conclusion and explanation)</b> ... (omitted PGM text analysis) ... Player 1 => It's sweet and juicy. Player 1: [[0.233 <b>0.384</b> 0.384]] Player 3: [[0.333 0.333 0.333]] [[0.333 0.333 0.333]] [[0.326 0.337 0.337]] [[0.333 0.333 0.333]] [[0.372 <b>0.391</b> 0.237]] Player 2 => It's often used in pies. Player 1: [[0.297 <b>0.358</b> 0.345]] Player 3: [[0.33 0.341 0.33]] [[0.326 0.337 0.337]] [[0.324 0.338 0.338]] [[0.322 0.345 0.333]] [[0.333 <b>0.366</b> 0.302]] Player 3 => It's tropical. Player 1: [[0.321 <b>0.354</b> 0.325]] Player 3: [[0.332 0.347 0.321]] [[0.327 0.342 0.331]] [[0.334 0.339 0.327]] [[0.319 0.348 0.333]] [[0.321 <b>0.356</b> 0.322]] Moderator => Now vote which of the other players (excluding yourself) is the chameleon. Player 1 => 1 vote for Player 2. Player 2 => 1 vote for Player 3. Player 3 => 1 vote for Player 2. Moderator => <b>The accusation is correct!</b> Player 2 is the chameleon! Now Player 2 can guess the secret code. Player 2 => I guess the code is "Pineapple". Moderator => <b>Player 2 guessed the code wrong! Non-Chameleon won!</b>	

Figure 5: A case study on Chameleon, Llama-2-70B, GPT-4, and their PGM-enhanced versions.

are playing rationally, the scores and payback will be much lower, thus approaching the well-known Nash Equilibrium (Kreps, 1989). In the Prisoner's Dilemma, if we compare GPT-3.5+PGM and GPT-4+PGM, GPT-4+PGM won more but got lower scores, showing that GPT-4+PGM made more rational decisions than GPT-3.5-turbo+PGM. In Public Good, we found models with PGM all achieved higher Win Rates but lower payback because they all performed more rationally in this scenario. If we compare the payback within models with or without PGMs, we can observe higher payback for GPT-4 models, which proves that GPT-4 models are more strategic in these games.

## 6 Discussion: Generalization of Benchmark

Beyond the scenarios mentioned in this paper, our benchmark can be generalized to more scenarios or tasks. In general, it evaluates agent behaviors in settings where the participants of a multi-agent system are usually local-viewed and need abilities involving Cognition, Adaptability, Rationality, and Collaboration to deduce the global information and make decisions, thus achieving the final goal. **Judgment** evaluates an agent's ability to accurately assess unknown information, such as roles in games like 'Chameleon' and 'Undercover'. **Reasoning** checks if an agent's perspective aligns with the actual and others' views, offering a nuanced understanding. **Self-awareness** adapts to scenarios

with undisclosed roles, while **Deception** looks at how well an agent can influence others with false information. **Cooperation** and **Coordination** gauge the effectiveness of collaborative efforts, measuring agreement and the quality of proposals, respectively. Lastly, we introduce **Rationality** from game theory, defining it as the proportion of decisions that maximize an agent's outcomes.

## 7 Conclusion

Our research presents a benchmarking framework tailored for evaluating LLMs in multi-agent environments. This framework's incorporation of diverse scenarios has enabled a quantitative assessment of seven critical abilities for LLMs in multi-agent systems, including judgment, reasoning, deception, self-awareness, cooperation, coordination, and rationality. The integration of PGM enriches LLMs with structural reasoning ability in multi-agent scenarios. Our quantitative analysis of 7 different multi-agent systems powered by various LLMs, including GPT-4-turbo, GPT-4, GPT-3.5-turbo, PaLM 2, Claude 2, Cohere, and Llama2-70B, has revealed their capabilities' disparity. Notably, GPT-4-turbo still emerged as the most capable, outperforming others by a threefold margin. Moreover, the PGM enhancement amplifies the inherent abilities of these models by 37%. This shows our benchmark's effectiveness and PGM's potential to enhance LLM capabilities.



## 632 Limitation

633 This paper proposed a benchmark for measuring  
634 the ability of LLMs in multi-agent systems. We  
635 conclude our limitation as below: Firstly, our in-  
636 vestigation of LLMs in multi-agent settings is in  
637 its preliminary stage. The scope of game scenar-  
638 ios and topic settings needs to be significantly ex-  
639 panded. Secondly, the PGM-aware method has  
640 the potential to enhance LLMs’ capabilities in the  
641 face of complex multi-agent settings. However, the  
642 process of integrating these incremental abilities  
643 into LLMs through methods such as fine-tuning  
644 requires further exploration. Third, whether the  
645 proposed metrics help in some of the real-life sce-  
646 narios is not explored in this paper.

## 647 Ethical Considerations

648 Our work introduces 5 multi-agent scenarios to  
649 evaluate LLMs. Most of the processes are under  
650 control with suitable prompts, so there is a low  
651 possibility of producing offensive outputs. We also  
652 checked all the data we use to ensure no personal  
653 data and unsuitable content included. All the data  
654 used in our scenarios are collected from public  
655 resources or generated by ChatGPT. We strictly  
656 follow the license for using the scientific artifact.

## 657 References

658 Sahar Abdelnabi, Amr Goma, Sarath Sivaprasad, Lea  
659 Schönherr, and Mario Fritz. 2023. Llm-deliberation:  
660 Evaluating llms with interactive multi-agent negotia-  
661 tion games. *arXiv preprint arXiv:2309.17234*.

662 Saaket Agashe, Yue Fan, and Xin Eric Wang. 2023.  
663 Evaluating multi-agent coordination abilities in large  
664 language models. *arXiv preprint arXiv:2310.03903*.

665 Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin John-  
666 son, Dmitry Lepikhin, Alexandre Passos, Siamak  
667 Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng  
668 Chen, Eric Chu, Jonathan H. Clark, Laurent El  
669 Shafey, Yanping Huang, Kathy Meier-Hellstern, Gau-  
670 rav Mishra, Erica Moreira, Mark Omernick, Kevin  
671 Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao,  
672 Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez  
673 Abrego, Junwhan Ahn, Jacob Austin, Paul Barham,  
674 Jan Botha, James Bradbury, Siddhartha Brahma,  
675 Kevin Brooks, Michele Catasta, Yong Cheng, Colin  
676 Cherry, Christopher A. Choquette-Choo, Aakanksha  
677 Chowdhery, Clément Crepy, Shachi Dave, Mostafa  
678 Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz,  
679 Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu  
680 Feng, Vlad Fienber, Markus Freitag, Xavier Gar-  
681 cia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-  
682 Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua

Howland, Andrea Hu, Jeffrey Hui, Jeremy Hur- 683  
witz, Michael Isard, Abe Ittycheriah, Matthew Jagiel- 684  
ski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, 685  
Sneha Kudugunta, Chang Lan, Katherine Lee, Ben- 686  
jamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, 687  
Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, 688  
Frederick Liu, Marcello Maggioni, Aroma Mahendru, 689  
Joshua Maynez, Vedant Misra, Maysam Moussalem, 690  
Zachary Nado, John Nham, Eric Ni, Andrew Nys- 691  
trom, Alicia Parrish, Marie Pellat, Martin Polacek, 692  
Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, 693  
Bryan Richter, Parker Riley, Alex Castro Ros, Au- 694  
rko Roy, Brennan Saeta, Rajkumar Samuel, Renee 695  
Shelby, Ambrose Slone, Daniel Smilkov, David R. 696  
So, Daniel Sohn, Simon Tokumine, Dasha Valter, 697  
Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, 698  
Pidong Wang, Zirui Wang, Tao Wang, John Wiet- 699  
ing, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting 700  
Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven 701  
Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav 702  
Petrov, and Yonghui Wu. 2023. Palm 2 technical 703  
report. *arXiv preprint arXiv:2305.10403*. 704

Anthropic. 2023. *Claude 2*. 705

Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. 706  
2011. Bayesian theory of mind: Modeling joint 707  
belief-desire attribution. In *Proceedings of the an- 708  
nual meeting of the cognitive science society*, vol- 709  
ume 33. 710

Maciej Besta, Nils Blach, Ales Kubicek, Robert Ger- 711  
stenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz 712  
Lehmann, Michal Podstawski, Hubert Niewiadomski, 713  
Piotr Nyczyk, et al. 2023. Graph of thoughts: Solv- 714  
ing elaborate problems with large language models. 715  
*arXiv preprint arXiv:2308.09687*. 716

Cohere. 2023. *Cohere for ai*. 717

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 718  
2023. Improving language model negotiation with 719  
self-play and in-context learning from ai feedback. 720  
*arXiv preprint arXiv:2305.10142*. 721

Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, 722  
Kiril Gashteovski, David Friede, Roberto Bifulco, 723  
and Carolin Lawrence. 2024. Agentquest: A modular 724  
benchmark framework to measure progress and im- 725  
prove llm agents. *arXiv preprint arXiv:2404.06411*. 726

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, 727  
Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. 728  
Reasoning with language model is planning with 729  
world model. *arXiv preprint arXiv:2305.14992*. 730

Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, 731  
Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, 732  
Xing Wang, Zhaopeng Tu, and Michael R Lyu. 2024. 733  
How far are we on the decision-making of llms? eval- 734  
uating llms’ gaming ability in multi-agent environ- 735  
ments. *arXiv preprint arXiv:2403.11807*. 736

Daphne Koller and Nir Friedman. 2009. *Probabilistic 737  
graphical models: principles and techniques*. MIT 738  
press. 739

740	David M Kreps. 1989. Nash equilibrium. In <i>Game Theory</i> , pages 167–177. Springer.	
741		
742	Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for "mind" exploration of large scale language model society. <i>arXiv preprint arXiv:2303.17760</i> .	
743		
744		
745		
746		
747	Minghao Li, Feifan Song, Bowen Yu, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. 2023b. Apibank: A benchmark for tool-augmented llms. <i>arXiv preprint arXiv:2304.08244</i> .	
748		
749		
750		
751	Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. <i>arXiv preprint arXiv:2308.03688</i> .	
752		
753		
754		
755	Roger B Myerson. 1991. <i>Game theory: analysis of conflict</i> . Harvard university press.	
756		
757	Ini Oguntola, Joseph Campbell, Simon Stepputtis, and Katia Sycara. 2023. Theory of mind as intrinsic motivation for multi-agent reinforcement learning. <i>arXiv preprint arXiv:2307.01158</i> .	
758		
759		
760		
761	OpenAI. 2023a. <a href="#">Gpt-3.5 turbo: A high-performance language model</a> . Whitepaper.	
762		
763	OpenAI. 2023b. <a href="#">Gpt-4 technical report</a> . <i>Preprint</i> , arXiv:2303.08774.	
764		
765	Joon Sung Park, Joseph C O’Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. <i>arXiv preprint arXiv:2304.03442</i> .	
766		
767		
768		
769		
770	Steve Phelps and Yvan I Russell. 2023. Investigating emergent goal-like behaviour in large language models using experimental economics. <i>arXiv preprint arXiv:2305.07970</i> .	
771		
772		
773		
774	Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. <i>arXiv preprint arXiv:2307.16789</i> .	
775		
776		
777		
778		
779	Toran Bruce Richards. 2023. Auto-gpt: An autonomous gpt-4 experiment.	
780		
781	Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. <i>arXiv preprint arXiv:2302.04761</i> .	
782		
783		
784		
785		
786	Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. <i>arXiv preprint arXiv:2303.11366</i> , 14.	
787		
788		
789		
790		
	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. <i>arXiv preprint arXiv:2307.09288</i> .	791
		792
		793
		794
		795
		796
	Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. <i>arXiv preprint arXiv:2305.16291</i> .	797
		798
		799
		800
		801
	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. <i>Advances in Neural Information Processing Systems</i> , 35:24824–24837.	802
		803
		804
		805
		806
	Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. 2023. Smartplay: A benchmark for llms as intelligent agents. <i>arXiv preprint arXiv:2310.01557</i> .	807
		808
		809
	S Yao, J Zhao, D Yu, N Du, I Shafran, K Narasimhan, and Y Cao. React: Synergizing reasoning and acting in language models. arxiv 2022. <i>arXiv preprint arXiv:2210.03629</i> .	810
		811
		812
		813
	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. <i>arXiv preprint arXiv:2305.10601</i> .	814
		815
		816
		817
		818
	Yao Yao, Zuchao Li, and Hai Zhao. 2023b. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. <i>arXiv preprint arXiv:2305.16582</i> .	819
		820
		821
		822
	<b>A Appendix</b>	823
	<b>A.1 Competition Settings</b>	824
	<b>Setting Definition</b> In Chameleon and Undercover, there are clearly two opposite roles, the Chameleon versus Non-Chameleons and the Undercover versus Civilians. The <i>challenger LLM</i> will play each role. For example, the <i>challenger LLM</i> plays non-chameleons versus GPT-4 as the chameleon, and the <i>challenger LLM</i> plays the chameleon versus GPT-4 as non-chameleons. The win rates of the <i>challenger LLM</i> playing different roles will be calculated separately, which contributes to in total 4 win rates.	825
		826
		827
		828
		829
		830
		831
		832
		833
		834
		835
	Cost Sharing has no distinct parties. Therefore, we made the <i>challenger LLM</i> as one player to play with other GPT-4-powered players. The final ratio of successful negotiations is defined as the win rate, which measures how much the LLM contributes to the agreement when other players are fixed. Similarly, for public good and prisoners’ dilemma, we	836
		837
		838
		839
		840
		841
		842

also made the *challenger LLM* as one of the players and recorded its win rate in these two games. The detailed win rate calculations are presented in A.2.

As shown in Table 3, we present the number of settings, corresponding metrics, and setting samples for each scenario. We build 20 settings for chameleon and undercover, respectively. In each game, Chameleon includes one round of clue giving while undercover contains 2 rounds. For each of the game theory scenarios, we collected 21 settings.

**Collection Process** In the Chameleon and Undercover scenarios, we’ve noticed a consistent bias in competition outcomes. Specifically, the Chameleon team has held an advantage in Chameleon, whereas in Undercover, the civilians have tended to win. To rectify this imbalance, we carried out 200 game simulations involving all three players as GPT-4 with randomly chosen topic settings. Through these simulations, we pinpointed 20 topic settings that promote a more equitable win rate between the two roles in both Chameleon and Undercover. In these scenarios, the challenger LLM will play both roles to measure different abilities such as judgment and deception, etc.

For the Cost-Sharing task, we expect all the participating airlines to share a fixed fee, with the specific share of each airline determined by its operational frequencies at the airport. These frequencies encompass various factors such as the number of flights, flight sizes, passenger volumes, and more. To facilitate the task, we asked ChatGPT to create a pool of 20 detailed descriptions of airline operational frequencies. A topic setting with 3 players is then constructed by three airline operational frequency descriptions from the pool, the role, and the position of the test LLM. Since there are 3 positions, we randomly selected 7 groups of airline operational frequency descriptions to form 21 distinct topic settings.

Similarly, for the two-game theory scenarios, we adopt a similar topic construction method as Cost Sharing. In the Prisoner scenario, three players choose to “defect” or “cooperate” for 5 rounds. Each player will get a different score depending on the outcomes of “defect” or “cooperate”. The player with the highest cumulative score wins the game. We have devised 7 distinct scoring settings, and the challenger LLM plays the role of each player across these settings, resulting in 21 unique competitions.

In the Public Good game, three players determine the number of points to contribute to a communal pool for 5 rounds. These invested points are multiplied by a specified factor (typically greater than 1), and the resulting sum is equally distributed among all players. Each player’s final score comprises their remaining points and the payback from the communal pool. The player achieving the highest score is declared the winner. We establish 7 different multipliers and assign the challenger LLM to play each of the three players in these settings, thus generating an additional 21 competitions.

## A.2 Win Rate Definition

In the chameleon, the outcome can be 0: the non-chameleon won, 1: the chameleon won, 2: even voting, and 3: the chameleon guessed right. In these four situations, credits gained by the role chameleon and non-chameleon are  $c_{chameleon} = [0, 1, 2, 1]$  and  $c_{non-chameleon} = [2, 1, 0, 1]$ , respectively. Suppose the outcomes of the  $n$  competitions are  $o$ . The total credits of all the completions are  $2n$ ; the win rate defined in Chameleon is

$$w_r = \frac{\sum_{i \in n} c_r[o_i]}{2n}, r \in [chameleon, non - chameleon]$$

Similarly, in Undercover, the outcome can be 0: undercover won, 1: civilian won, and 2: even voting. The credits for the role undercover and civilians are  $c_{undercover} = [3, 0, 2]$  and  $c_{civilian} = [0, 3, 1]$ , respectively.

$$w_r = \frac{\sum_{i \in n} c_r[o_i]}{2n}, r \in [undercover, civilian]$$

The win rate of cost sharing is the success rate of achieving consistency in all competition. In the game theory settings, the win rate is the ratio of the testing player winning the competition.

## A.3 PGM Enhancement Performance

We present all the experimental results in Table 4 and the corresponding radar chart for PGM-aware agents in Figure 6.

## A.4 More Case Studies

**Deception** Another advanced cognitive ability of LLMs extends to their proficiency in strategic deception within a multi-agent framework. In Figure 7, we delve into the dynamics of LLM performance when assuming an undercover role against GPT-4. In this scenario, LLMs are expected to

	Chameleon	Undercover	Cost Sharing	Prisoner's Dilemma	Public Good
Judgement			-	-	-
Reasoning			-	-	-
Deception			-	-	-
Self-Awareness			-	-	-
Collaboration	-	-		-	-
Coordination	-	-			
Rationality	-	-			
# Rounds	1	2	5	5	5
# Competitions	20	20	21	21	21
Setting sample	<pre>{   "topic": "Fruits",   "code": "Grape",   "chameleon_name": "Player 3",   "first clue": {     "Player 1": "It's often used to make wine.",     "Player 2": "It's small and round.",     "Player 3": "It's red in color."   } }</pre>	<pre>{   "undercover_code": "viewer",   "non_undercover_code": "listener",   "undercover_name": "Player 1",   "first clue": {     "Player 1": "My clue is: television.",     "Player 2": "My clue is: radio.",     "Player 3": "My clue is: audience."   } }</pre>	<pre>{   "test_player_name": "Player 3",   "topic": "Fixed Airport Fee: \$1,000,000   Airline Usage Frequency Data:   Airline A:   Number of Flights: 90/month   Number of Passengers: 7,000/month   Average Flight Duration: 1.75 hours   Flight Size: Primarily small and medium aircraft   Airline B:   Number of Flights: 50/month   Number of Passengers: 7,500/month/Average Flight Duration: 1.5 hours   Flight Size: Primarily small aircraft   Airline C:   Number of Flights: 50/month   Number of Passengers: 7,500/month   Average Flight Duration: 1.5 hours   Flight Size: Primarily small aircraft",   "first proposal": {     "Player 1": [40,30,30],     "Player 2": [50,25,25],     "Player 3": [50,25,25]   } }</pre>	<pre>{   "test_player_name": "Player 2",   "topic_values": {     "cooperate": 2,     "defect": 1,     "one_defect": 4,     "two_defect": 2   } }</pre>	<pre>{   "test_player_name": "Player 1",   "game_round": 5,   "multiplier": 3.5 }</pre>

Table 3: Consolidated Game Settings for Testing Abilities

	Win Rate	Judge.	Reason.	Decept.	Self-aware.	Cooper.	Coord.	Rational.
GPT-4-turbo+PGM	58.3	76.2	39.2	62.5	56.9	<b>81.0</b>	47.6	76.7
GPT-4-turbo	57.2	81.2	37.0	65.0	55.0	66.7	33.4	<b>78.1</b>
GPT-4+PGM	<b>63.5</b>	<b>87.5</b>	37.8	<b>75.0</b>	<b>61.3</b>	61.9	57.1	76.2
GPT-4	58.3	83.8	32.3	<b>75.0</b>	55.0	47.6	47.6	69.0
GPT-3.5-turbo+PGM	49.1	65.0	33.5	62.5	36.1	71.4	33.3	59.5
GPT-3.5-turbo	39.3	52.5	24.5	77.5	25.9	57.1	9.50	41.4
Claude 2 + PGM	43.0	57.5	<b>44.0</b>	42.5	60.0	85.7	<b>61.9</b>	54.8
Claude 2	34.0	45.0	34.0	25.0	50.0	71.4	23.8	24.3
PaLM 2 + PGM	41.4	62.5	39.3	60.0	34.5	42.9	4.80	40.0
PaLM 2	33.3	43.8	25.8	32.5	41.1	42.9	14.3	38.1
Cohere + PGM	35.8	52.5	31.8	67.5	30.4	42.9	4.80	30.0
Cohere	27.3	42.5	27.8	37.5	35.6	71.4	4.80	18.1
Llama-2-70B+PGM	30.8	53.7	29.3	55.0	45.2	52.4	14.3	28.1
Llama-2-70B	26.5	45.0	37.0	40.0	53.2	42.9	4.80	5.20
Average improvement	6.57	8.72	5.21	6.07	0.66	5.46	12.2	13.0

Table 4: Ability Measurement of LLMs.

blend in with regular civilians and even give misleading clues to conceal their actual roles. In this example, GPT-3.5-turbo, GPT-3.5-turbo+PGM lost the game, GPT-4 ended with even voting, and GPT-4+PGM won the game. According to their clues, we found models without PGM didn't tend to deceive others, and their clues describe their own words. Within these models, GPT-4 is more cautious when giving clues, while GPT-3.5 often gives very straightforward clues, like "It can be done at

a salon or barbershop" and "It can be washed with shampoo" to describe "hair cut".

In contrast, models augmented with PGM adopted a more sophisticated strategy, utilizing PGM analyses to decide their roles, identify potential undercovers, and outline strategic moves for subsequent rounds. This strategic advantage allowed these models to give fake clues to disguise themselves. For instance, as highlighted in Figure 7, "Player 2" portrayed by GPT-3.5-



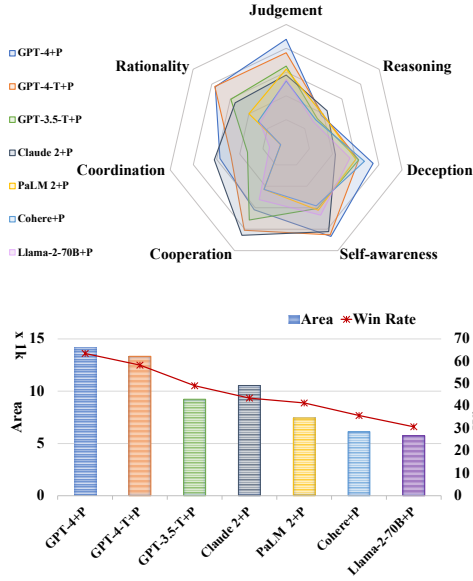


Figure 6: The radar chart depicts LLMs’ performance on 7 metrics of PGM-aware agents, with “-P” for “-PGM”. The bar chart displays the polygons’ areas, and the red line indicates average game-winning rates. Larger areas correlate with higher winning rates, validating the effectiveness of the proposed metrics for assessing LLMs’ capabilities.

turbo+PGM and GPT-4+PGM introduced deceptive clues, such as "It can be made from human and synthetic hair" and "It can be used to cover baldness," respectively. However, “Player 2” played by GPT-3.5-turbo+PGM, still lost the competition because its first clue looked like copying from “Player 1”. “Player 2” played by GPT-4+PGM, won the competition because of its consistently good clues in two rounds. Comparing the results of the three LLMs with PGM, we can conclude that a more capable LLM can benefit more from the PGM structure.

### A.5 PGM prompts

We have designed different prompts to test LLMs’s ability to make PGM analyses. Two kinds of prompts are used: text-based(**w.PGM**) and direct probability matrix(**w.ProPGM**). In specific, the former lets the LLM analyze global information in text, as shown by the example in Figure 3. The latter requires the LLM to directly give a probabilistic matrix to represent the global information, for example, a matrix  $[[0.3,0.2,0.5],[0.1,0.4,0.5],[0.3,0.3,0.4]]$ . The three roles represent  $B_1, B_2$ , and  $B_3$  respectively. Each element in a role is the probability of a player being the undercover or the chameleon for example. We compare the vote accuracy and Win rate of these

two kinds of prompts in the scenario chameleon, as shown in Figure 8. We found that more capable LLMs, like GPT-4 and GPT-3.5-turbo, both kinds of prompts work well. However, for Llama-2-70B the text PGM analysis performs much better. Therefore, we mainly choose the prompt the LLMs to give text-based PGM analysis to ensure the help of PGM on all the LLMs.

### A.6 Defect and Investment Tendency

**LLM awareness of arithmetic.** The upper diagram in Figure 9 shows the total investment amount under different payback multipliers. The x-axis is the 7 multipliers used in our topic settings. We told all the participating LLMs in a competition that the total amount of money they can invest is 100. However, except for the model GPT-4, the average amount of total investments of these LLMs almost all exceed 100, which indicates a lack of arithmetic awareness for these LLMs. In contrast, GPT-4 better understands the game rules and masters arithmetic in giving reasonable investment.

**LLM behaviors with varying topic settings.** To also investigate whether the LLMs will behave differently when given different topic settings. For example, in Public Good, we chose the multipliers  $[1,1.2,1.5,1.8,2.2,5,3]$ . We can find in the upper diagram in Figure 9, that even though the multiplier is increasing, all the lines do not show a tendency to increase investment, which proves that all the tested challenger LLMs are not sensitive to the increased possible payback in the communal pool in our current setting. Similar phenomena also happen in Prisoner’s Dilemma. As shown by the bottom diagram in Figure 9, the x-axis is the score criteria, which consists of three digits, representing the score a player can get in situations where the player defects when all the other two defect, the player defects when the other two cooperate, and the player defects when one of the other two defects, respectively.

### A.7 Game rules and Prompt Design

We have listed the game rules for our proposed 5 scenarios in Table 5 and 6.

GPT-3.5-turbo and GPT-4 as Undercover	
Moderator => Now the game starts! Moderator => [Player 2 ]Your word is: haircut ; Moderator => [Player 1 ]Your word is: wig; Moderator => [Player 3 ]Your word is: wig	
<p><b>GPT-3.5-turbo</b></p> <p>Player 1 =&gt; It's something you wear on your head. Player 2 =&gt; It's something you often get done at a salon or barbershop. Player 3 =&gt; It can be used to change your appearance. Moderator =&gt; Now we starting another round of giving clue... Player 1 =&gt; It can be of different colors. Player 2 =&gt; It's something that involves cutting and styling your hair. Player 3 =&gt; It can be synthetic or natural. Player 1 =&gt; I vote for Player 2; Player 2 =&gt; I vote for Player 3...; Player 3 =&gt; I vote for Player 2. Moderator =&gt; The most-voted player is Player 2. The accusation is correct. Player 2 is the undercover! <b>Civilians won!</b></p>	<p><b>GPT-4</b></p> <p>Player 1 =&gt; It's something you wear on your head. Player 2 =&gt; It's something that can change your appearance. Player 3 =&gt; It's often used in theater productions. Moderator =&gt; Now we starting another round of giving... Player 1 =&gt; It can be made of synthetic materials. Player 2 =&gt; It's something that can be styled in different ways. Player 3 =&gt; It's often used by people who have lost their hair. Player 1 =&gt; I vote for Player 2; Player 2 =&gt; I vote for Player 3; Player 3 =&gt; I vote for Player 1. Moderator =&gt; There are <u>even votes</u>. The accusation does not stand. Player 2 is the undercover.</p>
<p><b>GPT-3.5-turbo+PGM</b></p> <p>Player 1 (clue) =&gt; It's something you wear on your head. Player 2 (clue) =&gt; It's something you wear on your head, similar to a hat. Player 3 (clue) =&gt; It can change your appearance Moderator =&gt; Now we starting another round... Player 1 (clue) =&gt; It's often used in theater. Player 2 (clue) =&gt; <b>It can be made from human or synthetic hair.</b> Player 3 (clue) =&gt; It can come in different colors and styles. Player 1 =&gt; I vote for Player 2; Player 2 =&gt; I vote for Player 3; Player 3 =&gt; I vote for Player 2. Moderator =&gt; The accusation is correct! Player 2 is the undercover! <b>Civilians won!</b></p> <p><i>Player 2 (pgm) =&gt; Based on the clues given, Player 1's word could be "wig" and Player 3's word could be "mask" ....As Player 2, your word is "haircut" which does not fit the clues given by Player 1 and Player 3. Therefore, <b>you can conclude that you are the undercover player in this round.</b></i>  <i>2. I think <b>Player 1 thinks Player 3</b> is the undercover because Player 3's ... is more vague and could apply to various items, while Player 1's ... a specific item.</i>  <i>3. I think <b>Player 3 thinks Player 1</b> is the undercover because Player 1's clue ... is more specific and indicates a ... while Player 3's clue ... could apply to multiple items.</i>  <i>4. ...I will try clue as "It can be made from human or synthetic hair," which is similar to both a hat and a haircut, but still different enough to not give away my identity ...</i></p>	<p><b>GPT-4+PGM</b></p> <p>Player 1 (clue) =&gt; It's something you wear on your head. Player 2 (clue) =&gt; It's something that can change your appearance. Player 3 (clue) =&gt; It's often used in theatrical performances. Moderator =&gt; Now we starting another round of giving clue... Player 1 (clue) =&gt; It can be made from synthetic materials. Player 2 (clue) =&gt; <b>It can be used to cover baldness.</b> Player 3 (clue) =&gt; It comes in different colors and styles. Player 1 =&gt; I vote for Player 3; Player 2 =&gt; I vote for Player 3; Player 3 =&gt; I vote for Player 1. Moderator =&gt; The most-voted player is Player 3. The accusation is incorrect. Player 2 is the undercover. <b>Undercover won!</b></p> <p><i>Player 2 (pgm) =&gt; 1. Based on the clues, it seems like Player 1 and Player 3 are describing a wig.... Comparing this to my word, "haircut", it's clear <b>that I am the undercover</b> in this game. ...</i>  <i>2. I think <b>Player 1 thinks Player 3</b> is the undercover, because Player 3's clue about theatrical performances doesn't align as well with the idea of something you wear on your head or something made from synthetic materials.</i>  <i>3. I think <b>Player 3 thinks Player 1</b> is the undercover, because Player 1's clue about synthetic materials doesn't align as well with the idea of something used in theatrical performances.</i>  <i>4. I think the word of other players may be "wig". I will try to give my next clue as "It can be used to cover baldness", which is similar to them to hide myself.</i></p>

Figure 7: A Undercover case study on GPT-3.5-turbo, GPT-4 and their PGM-enhanced version (\*+PGM).

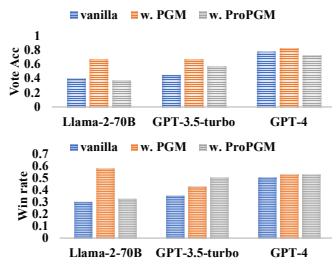


Figure 8: The performance comparison of different prompt designs in Chameleon. “ProPGM” refers to generating a PGM matrix directly with LLM. “PGM” is accumulated PGM extracted from text-based analysis.

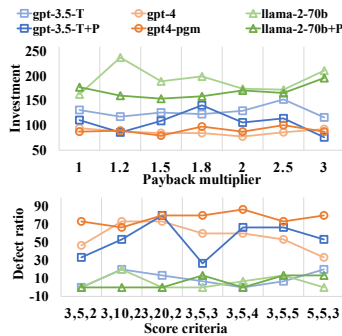


Figure 9: Features in game theory analysis.

Role	Global Prompt	PGM Prompt	PGM Decision Prompt
Chameleon	<p>You are playing a game of the Chameleon. Here are the game rules:</p> <p>## Information and roles</p> <p>There are two roles in the game, chameleon and non-chameleon. The topic of the secret word will be first revealed to all the players. Then the secret word will be revealed to non-chameleons. The chameleon does not know the secret word.</p> <p>## Objectives</p> <p>Your objective in the game depends on your role:- If you are not a chameleon, your goal is to reveal the chameleon without exposing the secret word.- If you are a chameleon, you aim to blend in with other players, avoid being caught, and figure out the secret word.</p> <p>## Stages</p> <p>There are three stages in the game:</p> <ol style="list-style-type: none"> <li>1. The giving clues stage: each player will describe the clues about the secret word.</li> <li>2. The accusation stage: In this stage, each player will vote for another player who is most likely the chameleon. The chameleon should vote for other players.</li> <li>3. The guess stage: If the accusation is correct, the chameleon should guess the secret word given the clues revealed by other players.</li> </ol>	<p><b>Non-Chameleon</b></p> <p>{player}, based on all the given clues, evaluate who is more suspicious of being a chameleon from your thoughts and the thoughts of other players you think. If you haven't seen a player's clue, don't think up one for him. You must follow the following format As {player}, I think: {other_player_1} is ..., because ... {other_player_2} is ..., because ... .. As for other players' thoughts: I think now {other_player_1} thinks: {player} is ..., because ... {other_player_2} is ..., because ... .. I think now {other_player_2} thinks: {player} ..., because ... {other_player_1} ..., because ... .. <b>Chameleon</b> {player}, deduce the secret code based only on {other_player_1} and {other_player_2}'s clues and analyze who is more suspicious from the perspective of other players based on all the clues given.</p>	<p><b>Non-chameleon:</b> {player}, you are a non-chameleon, your target is to increase the suspiciousness of the {target_player}. For example, by giving more specific clues than the {target_player}. Remember don't reveal the secret code;</p> <p><b>Chameleon:</b> {player}, you are a chameleon, and your target is to blend in and defraud trust from the {target_player}, try to describe the same word as other players.</p> <p>-If you are the first clue giver, try to be general in case you expose yourself.</p> <p>-If you guessed the secret code, try to give a specific clue about the word.</p> <p>-If other players' clues are general, try to give a clue similar to the {target_player}'s.</p>
Undercover	<p>You are playing a game of the Undercover. Here are the game rules:</p> <p>## Information and roles</p> <p>There are two roles in the game, undercover and non-undercover. Each Player will get one word but the one undercover gets will be different from the other players'.</p> <p>## Objectives</p> <p>First, you should guess who is undercover based on everyone's clues. Don't give too specific clues in case expose your own role. After you have guessed your own role, your objective in the game depends on your role:- If you are not undercover, your goal is to reveal the undercover without exposing your own word.- If you are undercover, your aim is to blend in with other players and avoid being caught.</p> <p>## Stages</p> <p>There are two stages in the game.</p> <ol style="list-style-type: none"> <li>1. The giving clues stage: each player will describe the clues in their own words. This stage will continue for several rounds.</li> <li>2. The accusation stage: In this stage, each player will vote for another player who is most likely undercover. The undercover should vote for other players.</li> </ol>	<p><b>Is undercover:</b></p> <p>You are {player}, your word is {code}, and guess the words other players are describing. Compare your word with other players' clues, if your word is different from what others are describing (even sometimes related), you may be the undercover. Do you think you are the undercover?</p> <p><b>Undercover:</b></p> <p>"{idx}. You are {player}, your word is {code}, and guess who is more likely to be the undercover from your own perspective. (hint: use the format, I think Player xx is undercover, because...)",</p> <p>"{idx}. You are {player}, your word is {code}. Now try to guess who {other_player} thinks is the undercover. (hint: use the format, I think {other_player} thinks Player xx is undercover, because...)",</p> <p>{idx}. I will try to a more specific clue about my word as... to help identify who is undercover."</p> <p><b>Civilian:</b></p> <p>{idx}. You are {player}, your word is {code}, and guess who is more likely to be the undercover from your own perspective. (hint: use the format, I think Player xx is undercover, because...),</p> <p>{idx}. You are {player}, your word is {code}, and you think you are undercover. Now try to guess who {other_player} thinks is the undercover. (hint: use the format, I think {other_player} thinks Player xx is undercover, because...),</p> <p>{idx}. I think the word of the other player may be ... I will try to give my next clue as ..., similar to hiding myself.</p>	<p><b>Not Sure:</b> {player}, according to your previous analysis, you are not sure who is undercover, give a not-too-specific clue, and don't reveal the word you have.</p> <p><b>Undercover:</b> {player}, according to your previous analysis, you think you are the undercover, your target is to blend in and defraud trust from the {target_player}.</p> <p>- If you have guessed other players' word, try to give a clue describing others' word;</p> <p>- If you are unsure about others' word, try to give clue similar to the {target_player}'s</p> <p><b>Civilian:</b> {player} according to your previous analysis, you think {target_player} is undercover. Try to increase {target_player}'s suspiciousness. For example, by giving more specific and different clues compared to the clue of {target_player}. You can give a clue based on your above analysis</p>

Table 5: Prompts for Undercover and Chameleon.

Role	Global Prompt	PGM Prompt	PGM Decision Prompt
Airport Fee	You are participating in the 'Airport Fee Allocation' game. You will each represent an airline and split the fixed cost of the airport. ## Information: - The total cost of the airport is fixed, and all airlines must collectively cover this cost. - Airlines have their unique usage frequencies at the airport, determined by factors like flight size and passenger volume. Airlines with higher usage frequencies are responsible for a larger portion of the cost. ## Objectives: - As an airline representative, your goal is to negotiate and agree upon a cost distribution that is both fair and favorable for your airline. ## Rules: The negotiation will continue for {max_turns} rounds. In each round: - Proposal: Each airline proposes a cost distribution. - Vote: Each player must vote for a cost distribution they find acceptable and strive to reach a consensus with other players' votes. - The game ends successfully when all airlines vote for the same proposal. If after {max_turns} rounds of voting no consensus is reached, the game fails. Strive for a successful outcome.	Please try to analyze how players will adjust their proposal from your perspective and other players in the next round. You must follow the template below ( make your comments concise): As {player_name}, I think {oth_player1} will... I think {oth_player2} will... As for other players, I think {oth_player1} thinks: {player_name} will... {oth_player2} will... I think {oth_player2} thinks {player_name} will...{oth_player1} will...	According to your own analysis of other players' possible decisions, try to adjust your proposal in the next round so that other players will all agree with you. - If some players stick to their decision and you think it is fair, you can follow their proposals to achieve agreement among you.
Prisoner's Dilemma	You are playing the Iterative Three-Player Prisoners' Dilemma game. Players 1, 2, and 3 are involved in a crime and are interrogated separately. They can either choose to betray the others (Defect, D) or cooperate with them (Cooperate, C). The rewards and penalties are distributed based on the combination of choices made by the three players. The scoring is as follows: - If all three choose to cooperate, they each get {cooperate} points. - If all three choose to defect, they each get {defect} point. - If one chooses to defect and the others choose to cooperate, the betrayer gets {one_defect} points and the collaborators each get 0 points. - If two choose to defect and the other chooses to cooperate, the betrayers each get {two_defect} points, and the collaborator gets 0 points. The game is played for 5 rounds, and the total score is calculated at the end to determine the winner. """"	Please try to analyze how other players will choose to defect or cooperate in the next round from your own perspective and others' perspectives. You must use the template: In my opinion: As {player_name}, I think {oth_player1} will xx, because... I think {oth_player2} will xx, because... As for other players, I think {oth_player1} thinks: {player_name} will... {oth_player2} will... I think {oth_player2} thinks {player_name} will... {oth_player1} will...	Based on your analysis of others and others' thoughts Remember to be strategic, establishing trust is to defect before others plan to do it to get the highest score. choose "defect" or "cooperate" to achieve the highest points for yourself. You must reply with "defect" or "cooperate"
Public Good	Welcome to the 'Public Good' game. Below are the game's settings and rules: - Each player begins with 100 monetary points. Over {game_round} rounds, you decide how many points to contribute to a communal pool at each round. - After {game_round} rounds of decision-making, the points in the communal pool will be multiplied by a factor of {multiplier} and distributed equally among all players. - A player's final points are the sum of their remaining points and the shared points from the communal pool. The player who gets the highest final points wins the game. - Every player must strategically invest their points to the communal pool to get more shared points and also be the one who invests the least to win the game. - Usually, you can get more payback by investing more when the factor is larger.	Please try to analyze whether other players will "reduce" or "increase" their contributions in the next round from your own perspective and others' perspective. - Remember, the payment in the communal pool is evenly shared by all players at the end of the game, so you need to make sure you invest the least money to get the highest repayment. You must use the template: In my opinion: As {player_name}, I think {oth_player1} will xx, because... I think {oth_player2} will xx, because... As for other players, I think {oth_player1} thinks: {player_name} will... {oth_player2} will... I think {oth_player2} thinks {player_name} will... {oth_player1} will...	Based on your analysis of others and others' thoughts, make the decision about your own contribution to achieving the highest repayment for yourself. Remember - Your total investment should be the least to win the game; - Your target is to get the highest points and also promote the communal repayment to get as many points as possible at the end. You must answer with the template "I contribute xx"

Table 6: Prompts in Cost-Sharing, Prisoner's Dilemma, and Public Good.