

# K-Level Reasoning: Establishing Higher Order Beliefs<sup>1</sup> in Large Language Models for Strategic Reasoning

Yadong Zhang<sup>1,2,\*</sup>, Shaoguang Mao<sup>2,†</sup>, Tao Ge<sup>2</sup>, Xun Wang<sup>2</sup>,<sup>2</sup>  
Yan Xia<sup>2</sup>, Man Lan<sup>1</sup>, Furu Wei<sup>2</sup>,

<sup>1</sup>East China Normal University, <sup>2</sup>Microsoft Research Asia<sup>3</sup>

## Abstract<sup>4</sup>

Strategic reasoning is a complex yet essential capability for intelligent agents. It requires Large Language Model (LLM) agents to adapt their strategies dynamically in multi-agent environments. Unlike static reasoning tasks, success in these contexts depends on anticipating other agents' beliefs and actions while continuously adjusting strategies to achieve individual goals. LLMs and LLM agents often struggle with strategic reasoning due to the absence of a reasoning framework that enables them to dynamically infer others' perspectives and adapt to changing environments. Inspired by the Level-K framework<sup>1</sup> from game theory and behavioral economics, which extends reasoning from simple reactions to structured strategic depth, we propose a novel framework: "**K-Level Reasoning with Large Language Models (K-R)**." This framework employs recursive mechanisms to enable LLMs to achieve varying levels of strategic depth, allowing agents to form higher order beliefs—beliefs about others' beliefs. We validate this framework through rigorous testing on four testbeds: two classical game theory problems and two social intelligence tasks. The results demonstrate the advantages of K-R in strategic reasoning. Our work presents the first recursive implementation of strategic depth in large language models (LLMs). It establishes a foundation for future research into theory of mind and strategic reasoning in LLMs.

## 1 Introduction<sup>6</sup>

Strategic reasoning—decision-making in multi-participant environments—presents unique challenges for Large Language Models (LLMs) and LLM agents(Zhang et al., 2024b). In these settings,

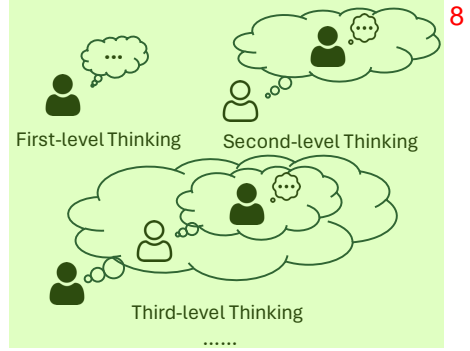


Figure 1: Level-K Framework: In first-level thinking,<sup>9</sup> agents respond directly to the environment. In second-level thinking, agents consider the first-level thinking of others. This process continues iteratively, with agents forming higher order beliefs based on assumptions about others' thoughts.

agents must respond to the actions of others while adapting to dynamic environments. They also need to align their decisions with their own goals during these interactions. Strategic reasoning is essential for intelligent agents and is widely applied in real-world tasks, such as investment, business strategy making(Zhao et al., 2023), negotiation(Hua et al., 2023), and policy-making(Li et al., 2024).

Effective strategic reasoning relies on understanding others' perspectives and anticipating their strategies. While there are some research efforts on LLMs' strategic reasoning, most methods rely on static prompting (Fu et al., 2023; Xu et al., 2023b). This typically involves instructing the model to account for others' beliefs and decisions during its own decision-making process in the prompt. However, these approaches fall short in enabling LLMs to form true higher order beliefs—beliefs about what others believe, and lack the flexibility needed for deeper strategic reasoning.

K-level thinking (Figure 1) (Nagel, 1995; Cui et al., 2021), a classical concept in behavioral economics and game theory, categorizes reasoning into varying depths of strategic thought. It involves not only predicting others' actions but also considering

\*Work was done when interning at Microsoft Research Asia. † Correspondence to: shaoguang.mao@microsoft.com

<sup>1</sup>According to the Level-k Framework, k-level thinking involves considering what opponent/partner are likely to do, what they think you will do, and what they believe you think they will do, and so on.

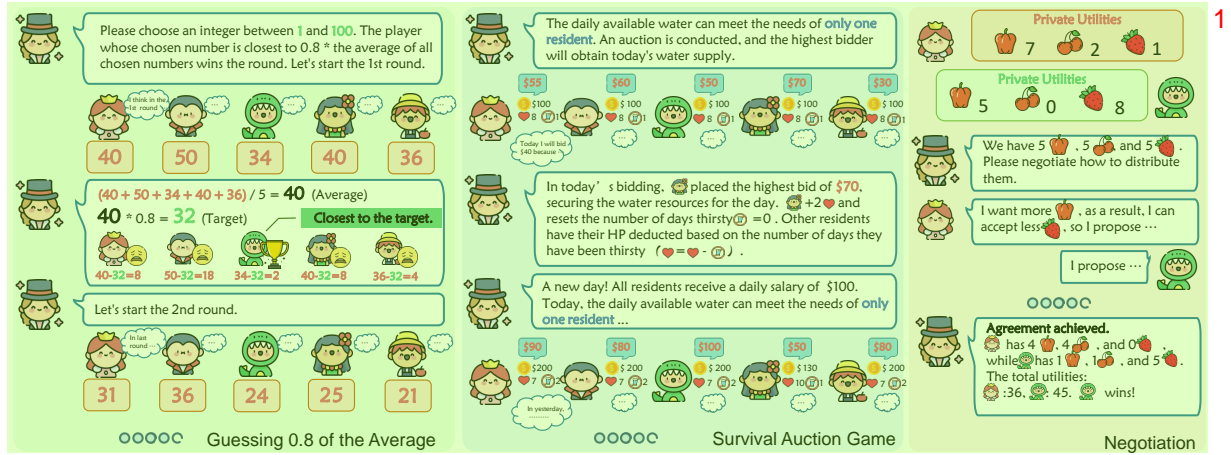


Figure 2: The illustration of three reasoning problems in dynamic, interactive environments in this paper. Left: Guessing 0.8 of the Average; Middle: Survival Auction Game; Right: Negotiation.

their beliefs about one's actions, and even further layers of recursive thinking.

Inspired by K-level thinking, we propose a novel strategic reasoning framework termed "K-Level Reasoning with LLMs (K-R)." K-R organizes reasoning into hierarchical levels and employs a recursive mechanism to integrate varying strategic depth into decision-making. Specifically, it involves: 1) recursively anticipating others' actions at varying levels of strategic depth with environmental context and historical public information, and 2) reasoning the optimal action based on these anticipations. To the best of our knowledge, this is the first approach to implementing varying levels of strategic depth in LLMs using a recursive mechanism and enables deeper reasoning in LLM agents through an algorithmic framework.

We validate this framework through rigorous testing on four testbeds: two classical game theory problems and two social intelligence tasks. The game theory problems includes Guessing 0.8 of the Average (Figure 2 left) and Survival Auction Game (Mao et al., 2023) (Figure 2 middle). The social intelligence tasks includes Negotiation (Cao et al., 2018) (Figure 2 right) and SOTOPIA benchmark (Zhou et al., 2024). These settings serve as microcosms of the complex decision-making processes involved in strategic reasoning. Through extensive experiments, we demonstrate that our framework significantly outperforms existing reasoning methods and flexibly achieves varying levels of strategic depth. In addition to empirical evidence, we provide a theoretical analysis highlighting the benefits of K-R. We show that, leveraging the in-context learning capabilities of LLMs, K-R

can effectively model opponents' behavior using accumulated public and available opponent information.

Furthermore, we align the strategic depth of LLMs with human participants (Nagel, 1995; Bosch-Domenech et al., 2002). Using human as anchors, we observe that the K-R significantly enhances the strategic depth of LLMs from 0.25 to 1.89. Notably, when K=3, the strategic depth (1.89) of the LLM closely approaches that of financial newspaper readers (1.91). This strongly indicates that K-R establishes higher order beliefs in LLMs for strategic reasoning.

The contributions of this work are as follows:

- We introduce K-R, a novel framework that extends k-level thinking to LLMs, enabling flexible strategic reasoning at varying depths through a recursive mechanism.
- We conduct extensive evaluations, including game theory and social intelligence problems, demonstrating that K-R significantly outperforms existing methods in terms of flexibility and effectiveness, across both closed-source and open-source models.
- We provide an in-depth analysis of K-R, confirming its ability to build higher order beliefs and enhance strategic reasoning. This lays a foundation for future research in theory of mind and strategic reasoning in LLMs.

## 2 K-Level Reasoning with Large Language Models

### 2.1 Methodology

Strategic reasoning requires considering both the decision context and the possible actions of other

participants. We employ a multi-round normal form multi-participant game to introduce the proposed method. In this setting, an agent's decision-making process is formalized as follows: each agent  $i$  selects an action  $a_i^t$  from a set  $A_i^t$  at timestep  $t$ . The payoff for agent  $i$ , resulting from the collective action profile  $\mathbf{A}^t = (a_1^t, a_2^t, \dots, a_N^t)$  and environment  $E^t$ , is denoted as  $U_i(E^t, \mathbf{A}^t)$ .

At  $k = 1$ , agents decide based on environment  $E^t$  without strategic anticipation:

$$a_i^{t,1} = \arg \max_{a_i \in A_i^t} \mathbb{E}[U_i(E^t, a_i)] \quad (1)$$

At higher level thinking ( $k \geq 2$ ), agent  $i$  simulates other agents operating at level  $k - 1$  and adjusts their strategy accordingly<sup>2</sup>:

$$a_i^{t,k} = \arg \max_{a_i \in A_i^t} \mathbb{E}[U_i(E^t, a_i, \hat{a}_{-i}^{t,k-1})] \quad (2)$$

where  $\hat{a}_{-i}^{t,k-1}$  are the predicted actions of other agents based on their  $k - 1$  level reasoning.

We propose a novel **strategic reasoning** framework with **recursive mechanisms**, termed “K-Level Reasoning with Large Language Models (K-R),” involving 1) recursively anticipating the actions  $\hat{a}_{-i}^{t,k}$  of others at different thinking levels using environment contexts and historical public information, followed by 2) reasoning the optimal action  $a_i^{t,k}$  based on anticipation of others' actions.

The K-Level Reasoning process is formulated as follows:

#### 1) Anticipation: 9

$$\hat{a}_j^{t,m} = \begin{cases} \text{LLM}(E^t, H_j^t) & \text{if } m = 1 \\ \text{LLM}(E^t, H_j^t, \hat{a}_{-j}^{t,m-1}) & \text{if } m > 1 \end{cases} \quad (3)$$

where  $\mathcal{H}_j^t = \{(E^1, a_j^1), (E^2, a_j^2), \dots, (E^{t-1}, a_j^{t-1})\}$  represents public historical data of agent  $j$ , and  $m$  denotes the specified thinking level.

#### 2) Reasoning: 12

$$a_i^{t,k} = \text{LLM}(E^t, H_i^t, \hat{a}_{-i}^{t,k-1}) \quad (4)$$

Algorithm 1 outlines the implementation of K-R. This recursive method enables flexible and progressively deeper strategic reasoning (1, 2, ...,  $k$ ,  $k + 1$ , ...), thereby enhancing higher order belief in LLM agents.

<sup>2</sup> To simplify the formulation, we assume that all opponents are in the same thinking level. In practice, varying thinking level can be implemented.

### Algorithm 1 K-Level Reasoning with LLMs

**Require:**  $E^t$ : Current decision context at time  $t$ ;  
 $H_i^t$ : Historical information up to time  $t$  for agent  $i$ ;  
 $K$ : Depth of strategic reasoning;  
**Ensure:**  $a_i^{t,K}$ : Action for agent  $i$  at time  $t$  after K-level reasoning.

- 1: **Function**  $K\_REASONING(i, k)$ :
- 2: **if**  $k == 1$  **then**
- 3:     **return**  $\text{LLM}(E^t, H_i^t)$
- 4: **else**
- 5:     **for each** agent  $j \neq i$  **do**
- 6:          $\hat{a}_j^{t,k-1} = K\_REASONING(j, k - 1)$
- 7:     **end for**
- 8:     **return**  $\text{LLM}(E^t, H_i^t, \{\hat{a}_j^{t,k-1} \mid j \neq i\})$
- 9: **end if**
- 10:  $a_i^{t,K} = K\_REASONING(i, K)$
- 11: **return**  $a_i^{t,K}$

## 2.2 Theoretical Analysis 16

This section discusses the benefits from K-R from a theoretical perspective. We utilize the in-context learning capabilities of LLMs to effectively model opponents' behavior. Suppose agent  $j$ 's decision-making process follows a hidden strategy  $\theta_j^*$ . Thus, agent  $j$ 's decision-making can be expressed as:

$$P(a_j^t \mid E^t, \theta_j^*) \quad (5)$$

The in-context learning of LLMs can be formally defined as implicit Bayesian inference (Xie et al., 2021); therefore, given the environment  $E^t$ , the next action prediction conditioned on  $H_j^t$  is:

$$P(a_j^t \mid E^t, H_j^t) = \int P(a_j^t \mid E^t, \theta_j) P(\theta_j \mid H_j^t) d\theta_j \quad (6)$$

As  $t \rightarrow \infty$ , by the law of large numbers and properties of Bayesian updating, the posterior distribution concentrates around the true parameter  $\theta_j^*$ :

$$P(\theta_j \mid H_j^t) \rightarrow \delta(\theta_j - \theta_j^*) \quad (7)$$

where  $\delta$  is the Dirac delta function. Therefore,

$$\int P(a_j^t \mid E^t, \theta_j) P(\theta_j \mid H_j^t) d\theta_j \rightarrow P(a_j^t \mid E^t, \theta_j^*) \quad (8)$$

This implies that as the number of interactions increases, K-R can more accurately predict opponents' behavior.

It is also worth noting that interaction data cannot be infinite, and in-context learning is related to

the performance of large language models (LLMs). Therefore, we empirically validate these hypotheses and reasoning in Section 5.2.

### 3 Experiments: Game Theory<sup>2</sup>

To fairly compare the strategic reasoning capabilities of LLMs, we first adopt two widely used game theory settings. These controlled, well-defined game theory problems provide a robust assessment of LLMs’ performance, with detailed setups outlined in Appendix B.

#### 3.1 Task Definition and Metrics<sup>4</sup>

##### 3.1.1 Guessing 0.8 of the Average (G0.8A)<sup>5</sup>

**G0.8A** (Figure 2 Left) is a classic game theory problem introduced by Alain Ledoux (Ledoux, 1981). It involves 10-round games where each player selects a number between 1 and 100. The objective is to choose a number closest to 80% of the group’s average choice. The key idea is to guess how others will estimate the average and decide the number to submit. This concept is also illustrated in the Keynesian Beauty Contest (Keynes, 1936). This game mirrors the challenge of anticipating collective behavior in financial markets, where investors must predict not only the value of an asset but also how others will value it in the future.

The performance of the agent is evaluated using the **Win Rate**. Specifically, the Win Rate is calculated based on the wins achieved by the agent in individual round, rather than an entire game episode.

##### 3.1.2 Survival Auction Game (SAG)<sup>15</sup>

**SAG** (Figure 2 Middle) is derived from the Water Allocation Challenge proposed in (Mao et al., 2023). Each resident’s goal is to survive a 10-day drought period by bidding for water resources and maintaining health points above zero. If a player successfully bids for water, they gain health points; otherwise, they lose health points. This integration of the auction system with the health points mechanism creates a dynamic environment where players must balance health and finances.

We use **Average Survival Round** measures the mean round in which a player remains active in the game.

#### 3.2 Base Techniques<sup>18</sup>

We adapt a variety of approaches, originally from traditional reasoning and agent benchmarks. These base techniques include:

**Standard Prompting (Direct)**: This is the conventional prompting method in which the LLM generates the final answer (Action) in response to the given game setting prompt.

**Chain-of-Thought (CoT)** (Wei et al., 2022): We employ the zero-shot Chain-of-Thought reasoning method (Kojima et al., 2022).

**Persona Prompting (Persona)** (Deshpande et al., 2023): This technique modifies the standard prompting process by incorporating a “Game Expert” persona to enhance the reasoning capabilities of the LLM.

**Reflexion (Reflect)** (Shinn et al., 2023): This method refers to language agents with verbal reinforcement learning and has been adapted for dynamic tasks. Detailed modifications are explained in K.

**Self-Refine (Refine)** (Madaan et al., 2023): This is a multi-round iterative reasoning approach where an additional LLM offers comments and adjustments prior to reaching a final decision. The distinctions between Self-Refine and Reflect are elaborated upon in the Appendix I.

**Prediction Chain of Thought (PCoT)**: This strong baseline diverges from CoT by requiring the LLM to explicitly predict opponents’ actions before making decisions. Unlike K-Level Reasoning, which involves a recursive approach, PCoT focuses on direct prediction based on context.

For implementation details and specific examples, please refer to Appendix K.

#### 3.3 Experimental Settings<sup>20</sup>

We established a controllable environment and distinguished between two roles: the player (primary focus) and the opponents. The player is equipped with a specific method, while all opponents use another reasoning approach. This well-defined setting allows for a clearer comparison of reasoning capabilities between methods.

In **G0.8A** and **SAG**, there is **one player** and **four opponents** for each game. Experiments for each setting are repeated 10 times and have passed the significance test (Appendix H), and each experiment consists of a 10-round game.

All methods in main experiments were implemented using GPT-4 (Achiam et al., 2023) (gpt4-32k), with the temperature set at 0.7 and the top-p set at 0.9. We also conducted experiments with open-source LLMs, Details of which are provided in Appendix E. Unless specified otherwise, the



level of thinking in K-Level Reasoning is set to K=2.

### 3.4 Results

To distinguish between “Player” and “Opponent” in the table, the headers for **Player** (bold) and *Opponents* (italics) are formatted accordingly.

Table 1: Win Rate of the player against different opponents in G0.8A game.

	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
<i>Direct</i>	0.43	0.67	0.62	0.53	0.43	0.61	<b>0.82</b>
<i>CoT</i>	0.07	0.32	0.35	0.14	0.22	0.45	<b>0.63</b>
<i>Persona</i>	0.05	0.37	0.29	0.05	0.37	0.11	<b>0.46</b>
<i>Reflect</i>	0.42	0.68	0.63	0.39	0.64	0.74	<b>0.78</b>
<i>Refine</i>	0.10	0.34	0.32	0.31	0.23	0.22	<b>0.46</b>
<i>PCoT</i>	0.03	0.44	0.52	0.21	0.51	0.54	<b>0.85</b>
<i>K-R</i>	0.04	0.15	0.14	0.04	0.17	0.14	<b>0.52</b>
Average	0.16 ± 0.18	0.32 ± 0.19	0.41 ± 0.18	0.24 ± 0.18	0.37 ± 0.17	0.40 ± 0.25	<b>0.65</b> ± 0.17

Table 2: Average Survival Round of the player against different opponents in Survival Auction Game.

	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
<i>Direct</i>	5.90	7.00	7.50	4.70	8.70	6.60	<b>9.40</b>
<i>CoT</i>	5.70	6.50	5.30	4.00	8.10	5.30	<b>10.00</b>
<i>Persona</i>	5.70	7.70	7.40	5.20	6.30	7.20	<b>9.30</b>
<i>Reflect</i>	9.40	9.40	9.90	5.20	8.60	8.20	<b>10.00</b>
<i>Refine</i>	6.30	6.40	8.10	4.30	<b>8.20</b>	5.30	7.90
<i>PCoT</i>	8.50	9.60	<b>9.90</b>	6.30	8.50	6.20	9.70
<i>K-R</i>	4.10	5.50	5.00	4.04	5.70	4.40	<b>6.80</b>
Average	6.51 ± 1.82	7.44 ± 1.55	7.59 ± 1.95	4.82 ± 0.82	7.73 ± 1.21	6.17 ± 1.29	<b>9.01</b> ± 1.21

Table 1 presents Win Rate of players utilizing different methods against various opponents in the **G0.8A** game. Notably, the K-R method demonstrates a superior Win Rate of 0.65, significantly exceeding the win rates of the other strategies. Table 2 provides insights into the Average Survival Round of players across different auction game strategies in **SAG**, with the K-R method again standing out. The K-R method achieves an average survival round of 9.01, considerably higher than all other methods.

The experiment result underscores the effectiveness of the K-R method in enhancing player strategy, suggesting its strategic superiority in the context of this game. Its effectiveness lies in its ability to anticipate opponent moves, outperforming other prompting methods.

The performance of Reflect did not demonstrate the effectiveness of the reasoning method. We hypothesize that this is due to the fact that, in dynamic

environments, Reflect on the experiences summarized from the previous round (Shinn et al., 2023) may not be applicable to the subsequent round of the game. Furthermore, in both games, Refine did not show an advantage over CoT and was significantly lower than K-R. This is because Refine involves adjustments based on one’s own strategy. However, these adjustments do not explicitly consider the hidden strategies of the opponent’s behavior, rendering them inapplicable against opponents employing different strategies.

## 4 Experiments: Social Intelligence

We then evaluate K-R in two social intelligence benchmarks to assess its performance in more open-ended realistic scenarios. Compared to the abstract and theoretical settings of Game Theory, these scenarios involve richer contextual backgrounds and complicated goal pursuits, which better demonstrate the value of LLM-based agents in practical applications, such as in chatbots and strategic decision making.

### 4.1 Task Definition and Metrics

#### 4.1.1 Negotiation (NEG)

NEG (Figure 2 Right)(Cao et al., 2018; Duan et al., 2024) is an open-ended and realistic task. In this setting, two agents are presented with three types of items: peppers, cherries, and strawberries. Each agent has private utility values for these items and must negotiate to allocate the public item pool.

The agent who secures more utility upon reaching an agreement wins the game, and we calculated the **Win Rate** to assess the performance of different agents.

#### 4.1.2 SOTOPIA Benchmark

**SOTOPIA** (Zhou et al., 2024) is an open-ended environment to simulate complex social interactions between artificial agents and evaluate their social intelligence. It includes a variety of social scenarios, and each scenario includes a context background, and private social goals of each agent. Meanwhile, each agent has a character profiles which consists of name, gender, personality, occupation, etc.

For each episode, agents are scored at the end of the interaction along each of seven dimensions in **SOTOPIA-Eval**, including Goal Completion (GOAL), Believability (BEL), Knowledge (KNO), Secret (SEC), Relationship (REL), Social Rules (SOC), Financial and Material Benefits (FIN).

## 4.2 Experimental Settings<sup>1</sup>

We employed the majority of the reasoning approaches introduced in Section 3.2 as baseline models for comparison.<sup>2</sup>

In **NEG**, the experiments followed the settings from (Cao et al., 2018; Duan et al., 2024). There is **one player** and **one opponent** for each game. We test the performance of the Baselines and K-Level Reasoning in 100 repeated independent games. To eliminate positional advantages, we swapped the positions of each player for each setting. To ensure the reliability, three trials were conducted, and the results are reported as averages with standard deviation.<sup>3</sup>

Meanwhile, We adhered to the **SOTOPIA-hard** (Zhou et al., 2024) setup comprising a total of 100 episodes, which is commonly found to be challenging for LLMs, and utilize a fixed GPT-4o based agent as partner. Additionally, to evaluate the agents' scores, we utilized GPT-4 as the assessment model, as it has been determined by SOTOPIA benchmark (Zhou et al., 2024) to serve as a reliable proxy for human judgments in evaluating model performance across most dimensions and for human performance on the GOAL dimension.<sup>4</sup>

## 4.3 Results<sup>5</sup>

Table 3: Win Rate of the player against opponent in Negotiation Setting.<sup>6</sup>

	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
Direct	<u>50.00</u>	61.34	49.58	66.67	65.83	63.03	<b>70.83</b>
CoT	38.66	<u>50.00</u>	36.67	45.83	45.76	47.27	<b>55.36</b>
Persona	50.42	63.33	<u>50.00</u>	70.00	67.50	62.50	<b>70.83</b>
Reflection	33.33	54.17	30.00	<u>50.00</u>	<b>57.14</b>	<u>55.00</u>	<u>55.00</u>
Refine	34.17	54.24	32.50	42.86	50.00	<b>55.77</b>	<u>54.55</u>
PCoT	36.97	52.73	37.50	45.00	44.23	<u>50.00</u>	<b>57.00</b>
K-R	29.17	44.64	29.17	45.00	<u>45.45</u>	43.00	<b>50.00</b>
Average	38.96 ±2.53	<u>54.35</u> ±0.50	37.92 ±5.84	52.19 ±1.73	53.70 ±4.41	53.80 ±4.34	<b>59.08</b> ± 2.20

The results presented in Table 3 and Table 4 illustrate the effectiveness of the K-Level Reasoning in the context of NEG and SOTOPIA-hard settings, respectively.<sup>8</sup>

In NEG, the K-R method demonstrates a notable win rate of 59.08%, positioning it significantly above the average win rates achieved by other methods. This indicates that, in most cases, the proposals generated through K-Level Reasoning are more advantageous to itself, as well as suggesting a tendency to accept the opponent's proposals when the

Table 4: SOTOPIA-Eval of the player against opponent in SOTOPIA-hard.<sup>10</sup>

Metric	Direct	CoT	Refine	K-R	Direct	CoT	Refine	K-R
	[GPT-4o]				[LLaMA-3.1-70B]			
BEL [0-10]	8.97	9.00	9.00	8.97	8.88	8.85	8.90	<b>8.97</b>
REL [-5-5]	2.38	2.40	2.27	<b>2.67</b>	1.38	1.18	0.82	<b>2.40</b>
KNO [0-10]	6.05	6.05	6.25	6.25	5.88	5.53	5.33	<b>6.12</b>
SEC [-10-0]	0.00	-0.05	0.00	0.00	-0.28	-0.25	-0.18	<b>0.00</b>
SOC [-10-0]	-0.05	0.00	-0.05	0.00	-0.70	-0.72	-0.64	<b>0.00</b>
FIN [-5-5]	<b>0.90</b>	0.78	0.80	0.72	0.38	0.35	-0.08	<b>0.75</b>
GOAL [0-10]	6.35	<b>6.60</b>	6.15	6.47	5.35	5.40	4.95	<b>6.38</b>
Overall	3.51 ± 0.09	3.54 ± 0.08	3.49 ± 0.08	<b>3.59</b> ± 0.09	2.98 ± 0.23	2.90 ± 0.26	2.73 ± 0.25	<b>3.52</b> ± 0.13

perceived benefits are substantial.<sup>12</sup>

The results from SOTOPIA reveal several intriguing findings. Firstly, while K-R demonstrates some improvement compared to other methods, the results are not statistically significant. We hypothesize that this may be due to the inherent tendency of GPT-4 based models to assign higher scores to responses generated by GPT-4 based agents. Notably, we observed that employing agents based on LLaMA 3.1 70B with K-R can lead to significant performance enhancements. Meanwhile, the overall metrics indicate that K-R achieves performance levels comparable to those of the GPT-4 model, highlighting K-R's potential in the realm of social intelligence.<sup>13</sup>

## 5 Discussions<sup>14</sup>

### 5.1 Does K-R Efficiently Establish a Higher Order Belief in LLMs?<sup>15</sup>

Table 5: Human performance in G2/3A.<sup>16</sup>

Experiments	Lab	Classroom	Take-home	Theorists	Internet Newsgroup	Newspaper
Mean Choice	35.13	26.84	25.20	17.15	22.16	23.08
Strategic Depth	0.87	1.53	1.68	<b>2.63</b>	2.01	1.91

Table 6: LLM performance in G0.8A in the first round.<sup>18</sup>

Method	Direct	CoT	Persona	Refine	Reflect	PCoT	KR[k=2]	KR[k=3]
Mean Choice	47.29	37.8	41.0	41.0	45.2	44.0	38.42	32.79
Strategic Depth	0.25	1.25	0.89	0.89	0.45	0.57	1.18	<b>1.89</b>

As a classic game theory issue, the G0.8A problem has garnered significant research interest across various disciplines. We reference the experimental results of the classic research among human participants (Nagel, 1995; Bosch-Domenech et al., 2002) as anchor points and present the average decisions made by the K-Level Reasoning method (GPT-4) in the first round. Through this

comparison, we can observe the relative relationship between human cognitive levels and LLMs under different reasoning methods. The specific calculation method on strategic depth is described in Appendix C. The performance of humans and LLMs is shown in the Table 5 and Table 6

From these observations, we can conclude that even when employing SOTA models, the strategic depth of GPT-4 under Direct Prompt (0.25) cannot compete with that of lower-strategic-capability undergraduate students in laboratory settings (0.87). Furthermore, the K-Level reasoning approach significantly enhances the reasoning depth of large language models, increasing it from 0.25 to 1.89, and the strategic depth of the large language model (1.89) approaches that of a group of financial newspaper readers (1.91) when  $K=3$ .

## 5.2 K-Level Reasoning Leads to More Accurate Predictions About Opponents

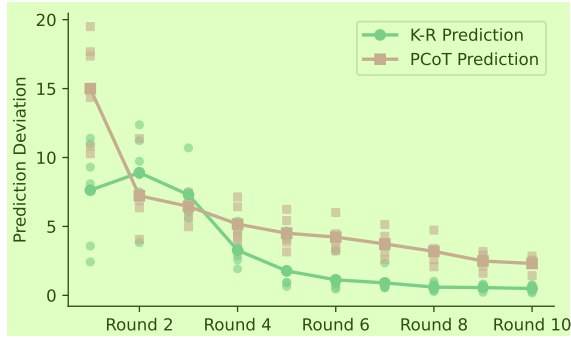


Figure 3: The deviation in prediction during the G0.8A between PCoT and K-Level Reasoning.

Since K-R involves an intermediate step of modeling the opponent’s behavior, we examine the progression of prediction accuracy. Figure 3 illustrates the prediction deviation between K-R and PCoT in G0.8A. K-R exhibits higher prediction accuracy than PCoT from Round 1, starting with more precise and less random predictions. Moreover, the predictions converge quickly and become highly accurate in the second half of the game. This trend highlights the LLM’s increasing proficiency in understanding higher order belief with more gameplay context. Essentially, K-R instantiates new sessions to compute the opponent’s future actions. This approach leverages the in-context learning capabilities of LLMs more effectively than PCoT’s prediction process (as theoretically discussed in Section 2.2). As a result, K-R achieves better prediction accuracy.

## 5.3 Better Reasoning Methodology vs. Stronger Foundation Model

There is a consensus that LLMs trained with more data and possessing larger parameter sizes demonstrate stronger reasoning capabilities. We explore whether K-Level Reasoning can significantly enhance the strategic reasoning abilities of relatively weaker LLMs. To investigate, we conducted experiments comparing the performance of K-R with GPT-3.5 (K-R[GPT-3.5]) against other reasoning methods based on GPT-4. All experiments were repeated 10 times.

Table 7: A comparison of K-Level Reasoning with GPT-3.5 and other reasoning approaches with GPT-4. For the Guessing 0.8 of the Average, we report the win rate; for the Survival Auction Game, we report the average survival round.

Opponent	Guessing 0.8 of the Average				Survival Auction Game			
	Direct [GPT-4]	K-R [GPT-3.5]	Direct [GPT-3.5]	K-R [GPT-4]	Direct [GPT-3.5]	K-R [GPT-3.5]	Direct [GPT-4]	K-R [GPT-4]
Direct	0.18	0.18	0.43	0.82	5.00	9.40	5.90	9.40
CoT	0.14	0.37	0.07	0.63	5.30	8.10	5.70	10.00
Persona	0.10	0.23	0.05	0.46	5.00	7.50	5.70	9.30
Reflect	0.24	0.38	0.42	0.78	5.00	8.50	9.40	10.00
Refine	0.14	0.13	0.10	0.46	5.10	6.70	6.30	7.90
PCoT	0.19	0.46	0.03	0.85	4.10	6.80	8.50	9.70
Average	0.16	0.29	0.18	0.67	4.92	7.83	6.92	9.38

From the results in Table 7, we observe that K-R[GPT-3.5] outperforms the standard prompting method of GPT-4 (Direct[GPT4]) from average performance. Furthermore, when competing against opponents using reasoning methods on GPT-4, K-R[GPT-3.5] demonstrates remarkable capabilities. K-R, with its excellent restoration of the rival’s perspective, enhances the LLM’s ability in competitive environments. Additionally, we compared the performance of the open-source model LLAMA2-7B with GPT-3.5/4 in Appendix E, finding that K-R significantly enhances reasoning in interactive contexts across different LLMs.

## 5.4 The Deeper Thinking Level, the Better Strategic Performance?

Table 8: Comparison between K-Level Reasoning[K=2] and K-Level Reasoning[K=3] in the two games.

Opponent	Guessing 0.8 of the Average			Survival Auction Game		
	Direct	K-R[K=2]	K-R[K=3]	Direct	K-R[K=2]	K-R[K=3]
Direct	0.43	0.82	0.77 (-0.05)	5.90	9.40	9.40 (+0.00)
K-R[K=2]	0.04	0.52	0.60 (+0.08)	4.10	6.80	8.30 (+1.50)

K-R models opponents’ thinking processes recursively. We examine how thinking level affect

reasoning outcomes by comparing K-R[K=2] and K-R[K=3] in two games. The results, detailed in Table 8, reveal the impact of increased thinking level. Against the Direct method (first-level thinking), K-R[K=3] showed a decreased win rate in G0.8A but maintained performance in SAG, suggesting possible overthinking. However, K-R[K=3] improved significantly against K-R[K=2] in both games. It suggests that the key factor in K-R is the relative depth of thought compared to the opponent. A one-level deeper approach offers a strategic advantage, but advancing two levels may lead to diminishing returns due to over-anticipation. In interactive environments, identifying opponents' thinking levels is difficult. Adapting to varying levels and using K-Level Reasoning for deeper analysis is a valuable direction for future research.

Additionally, a higher thinking level with the recursive prompting implementation increases computational cost. The computational cost of K-R is thoroughly discussed in Appendix G.

## 6 Related Work

### 6.1 Reasoning with LLMs

Large Language Models (LLMs) excel in diverse complex reasoning tasks, such as mathematical (Miao et al., 2021; Patel et al., 2021), common sense (Talmor et al., 2022; Bhakthavatsalam et al., 2021), and symbolic reasoning (Srivastava et al., 2022; Suzgun et al., 2022). A notable reasoning approach involves breaking down complex questions into a series of intermediate steps, a technique known as the Chain-of-Thought (CoT) method (Wei et al., 2022; Kojima et al., 2022). Subsequently, some works have emerged to extend CoT, with innovations like Tree of Thought (ToT) (Yao et al., 2023), Graph of Thought (GoT) (Besta et al., 2023) and Skeleton-of-thought (Ning et al., 2023). Besides, approaches like Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) enhance CoT's consistency by having LLMs review and refine their responses. Moreover, recent research has revealed that integrating persona information into LLMs significantly improves their reasoning processes (Deshpande et al., 2023). A series of studies (Fu et al., 2023; Wang et al., 2023) have been conducted to incorporate more persona information, aiming to enhance the rationality and knowledge ability of the LLM reasoning process. These methods have been applied to various static tasks, but have not been adequately evaluated in

dynamic problems (multi-agent environment) to validate their efficacy in reasoning capabilities.

### 6.2 Strategic Reasoning within Multiple Agent System

Dynamic problems arise when multiple participants are involved in multi-round interactions. One key factor is the simultaneous interactions of multiple participants with the environment. Unlike single-agent systems, multiple agent system (MAS) encounters a broader range of issues and challenges, as noted by (Wong et al., 2021), including computational complexity (Ding and Dong, 2020), nonstationarity (Papoudakis et al., 2019), partial observability (Mahajan et al., 2019; Foerster et al., 2016), and challenges in credit assignment (Sunehag et al., 2017). Particularly, in the context of inference using LLMs, the nonstationarity of the environment poses a distinct challenge.

Recently, research on LLMs in strategic reasoning has been conducted across various MAS including social behavior(Zhou et al., 2024; Hua et al., 2023), economic simulations(Zhao et al., 2023; Li et al., 2023), game theory(Duan et al., 2024; Xu et al., 2023a), and game playing(Ma et al., 2023; Xu et al., 2023b). To enhance the performance of LLMs in strategic reasoning scenarios, researchers have utilized the concepts of Theory of Mind (ToM) (Gandhi et al., 2023; Guo et al., 2023) and Reinforcement Learning (Xu et al., 2023c; Zhang et al., 2024a) to optimize the reasoning processes of LLMs. These approaches involve prompting LLMs to recognize the intricacies of strategic tasks, like our proposed Prediction Chain-of-Thought baseline. However, our experimental results indicate that this approach fails to establish a clear cognitive hierarchy necessary for recursive and deeper strategic thinking.

## 7 Conclusion

This paper represents a significant stride in understanding and enhancing the strategic reasoning capabilities of LLMs. We propose “K-Level Reasoning with LLMs.” This innovative approach leverages recursive mechanisms to achieve varying thinking level within LLMs, enabling them to engage in deeper strategic thinking. Through extensive experiments, we validate the advantage offered by this method. It establishes a foundation for future research into theory of mind and strategic reasoning in LLMs.



## 8 Limitations<sup>1</sup>

We validate the effectiveness of the K-Level Reasoning framework from two perspectives: game theory and social intelligence. While our experimental results provide substantial evidence supporting the framework’s validity, further research is necessary to explore the performance of large language models (LLMs) in few-shot agent modeling (He et al., 2016) across various environments, strategic factors, and action sets.

Additionally, K-R predicts opponents’ most likely behavior by initiating a new LLM inference session. The recursive mechanism employed to achieve varying levels of strategic depth inevitably increases computational cost. Appendix G provides a detailed discussion on how K-R relates to this rise in computational cost and compares it across different reasoning methods. Despite the increased demands, K-R outperforms other methods with comparable computational costs.

## References<sup>5</sup>

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge. *arXiv preprint arXiv:2102.03315*.

Antoni Bosch-Domenech, Jose G Montalvo, Rosemarie Nagel, and Albert Satorra. 2002. One, two,(three), infinity,...: Newspaper and lab beauty-contest experiments. *American Economic Review*, 92(5):1687–1701.

Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*.

Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Foerster. 2021. K-level reasoning for zero-shot coordination in hanabi. *Advances in Neural Information Processing Systems*, 34:8215–8228.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335*.

Zihan Ding and Hao Dong. 2020. Challenges of reinforcement learning. *Deep Reinforcement Learning: Fundamentals, Research and Applications*, pages 249–272.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations. *arXiv preprint arXiv:2402.12348*.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29.

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

Kanishk Gandhi, Dorsa Sadigh, and Noah D Goodman. 2023. Strategic reasoning with language models. *arXiv preprint arXiv:2305.19165*.

Jiaxian Guo, Bo Yang, Paul Yoo, Bill Yuchen Lin, Yusuke Iwasawa, and Yutaka Matsuo. 2023. Suspicion-agent: Playing imperfect information games with theory of mind aware gpt-4. *arXiv preprint arXiv:2309.17277*.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. 2016. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pages 1804–1813. PMLR.

Wenyue Hua, Lizhou Fan, Lingyao Li, Kai Mei, Jianchao Ji, Yingqiang Ge, Libby Hemphill, and Yongfeng Zhang. 2023. War and peace (waragent): Large language model-based multi-agent simulation of world wars. *arXiv preprint arXiv:2311.17227*.

John Maynard Keynes. 1936. The general theory of employment. *The quarterly journal of economics*, 51(2):209–223.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Alain Ledoux. 1981. Concours résultats complets. *Les victimes se sont plu à jouer le*, 14:10–11.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2024. EconAgent: Large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting*

- of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15523–15536, Bangkok, Thailand. Association for Computational Linguistics.
- Yang Li, Yangyang Yu, Haohang Li, Zhi Chen, and Khaldoun Khashanah. 2023. Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance. *arXiv preprint arXiv:2309.03736*.
- Weiyu Ma, Qirui Mi, Xue Yan, Yuqiao Wu, Runji Lin, Haifeng Zhang, and Jun Wang. 2023. Large language models play starcraft ii: Benchmarks and a chain of summarization approach. *arXiv preprint arXiv:2312.11865*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32.
- Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei. 2023. Alympics: Language agents meet game theory. *arXiv preprint arXiv:2311.03220*.
- Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.
- Rosemarie Nagel. 1995. Unraveling in guessing games: An experimental study. *The American economic review*, 85(5):1313–1326.
- Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *arXiv preprint arXiv:2307.15337*.
- Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. 2019. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.
- Dale O Stahl and Paul W Wilson. 1995. On players’ models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1):218–254.
- Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.
- Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2022. Commonsenseqa 2.0: Exposing the limits of ai through gamification. *arXiv preprint arXiv:2201.05320*.
- Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300*.
- 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. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Annie Wong, Thomas Bäck, Anna V Kononova, and Aske Plaat. 2021. Deep multiagent reinforcement learning: Challenges and directions. *arXiv preprint arXiv:2106.15691*.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.
- Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See-Kiong Ng, and Jiashi Feng. 2023a. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.
- Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023b. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*.

- Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2023c. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*.<sup>1</sup>
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.
- Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024a. Agent-pro: Learning to evolve via policy-level reflection and optimization. *arXiv preprint arXiv:2402.17574*.
- Yadong Zhang, Shaoguang Mao, Tao Ge, Xun Wang, Yan Xia, Wenshan Wu, Ting Song, Man Lan, and Furu Wei. 2024b. [LLM as a mastermind: A survey of strategic reasoning with large language models](#). In *First Conference on Language Modeling*.
- Qinlin Zhao, Jindong Wang, Yixuan Zhang, Yiqiao Jin, Kaijie Zhu, Hao Chen, and Xing Xie. 2023. Competeai: Understanding the competition behaviors in large language model-based agents. *arXiv preprint arXiv:2310.17512*.
- Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haoifei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024. [SOTOPIA: Interactive evaluation for social intelligence in language agents](#). In *The Twelfth International Conference on Learning Representations*.

## A Impact Statements<sup>1</sup>

Our research introduces the K-Level Reasoning framework, designed to formulate strategies in dynamic, interactive and competitive scenarios by anticipating the reactions of adversaries, potentially users. Theoretically, this approach offers a novel perspective for understanding and optimizing decision-making processes. However, we recognize when the goal setting diverges from user interests, the application of K-Level Reasoning could result in manipulative behaviors by adapting to the predicted user's reactions. This risk is notably pronounced in scenarios designed to influence user decisions or behaviors, such as in recommendation systems, advertising placements, and content distribution on social media platforms.

Although K-Level Reasoning provides a potent powerful tool for strategic planning, interacting and reasoning, ethical considerations must be meticulously managed in its practical application. This ensures that the development and utilization of technology do not detrimentally impact user and societal interests. To this end, we advocate for heightened transparency, ensuring users have a comprehensive understanding and control over how their data is utilized.

## B Game Setting<sup>4</sup>

### B.1 Guessing 0.8 of the Average<sup>5</sup>

**Initial Setup:** For each round, each player select a number between 1 and 100. The objective is to select a number that is closest to 80% of the group's average choice. Formally, each player  $i$  chooses a number  $n_i$ , aiming for  $n_i \approx 0.8 \times \bar{n}$ , where  $\bar{n}$  is the average of all chosen numbers.

**Scoring and Continuation:** A player scores a point if his/her chosen number is closest to 80% of the average number chosen by the group. If all players select the same number, no points are awarded for this round. Mathematically, the score for player  $i$  in in round  $t$  is given by  $s_i^t$ , which is 1 if  $|n_i - 0.8 \times \bar{n}|$  is the minimum among all players, and 0 otherwise.

### B.2 Survival Auction Game<sup>8</sup>

**Initial Setup:** Players start with 8 health points, out of a maximum of 10. Every day, each player possesses a fixed income of \$100. The daily water supply can only satisfy one resident's requirement.

**Scoring and Continuation:** Everyday, players engage in a daily auction to secure the necessary

water resources, and the highest bidder wins. In case of a tie, the resources are not allocated to any player. If a player successfully bid the water resources, they will gain 2 health points; otherwise, they will lose health points equal to the number of consecutive days, denoted as  $n$ , during which they have not obtained water resources. Once a player's health points fall to 0 or below, they will be eliminated. The health point of player  $i$  on day  $t$ , denoted as  $h_i^t$ , is crucial in determining their survival and bidding strategy.

## C Detailed Metric Computational Formulas<sup>12</sup>

**Win Rate** is calculated based on the number of wins over game going, providing a measure of the overall ability.

$$\text{WinRate} = \frac{\text{Num of Wins}}{\text{Total Round per Test} \times \text{Num of Test}} \quad (9)$$

**Average Survival Round** calculates the average round in which the player remains in the game. It's an effective way to assess performance in elimination-based game, like SAG.

$$\text{AvgSurvivalRound} = \frac{\sum \text{Survival Round in Each Test}}{\text{Num of Test}} \quad (10)$$

**Prediction Accuracy** evaluates the accuracy of player's predictions regarding rivals' future moves. In the G0.8A, it involves calculating the absolute difference between the player's predicted average and the actual average in each round:

$$\text{Pred Acc} = \frac{\sum |\text{Avg}_{\text{Pred}} - \text{Avg}_{\text{Actual}}|}{\text{Num of Test}} \quad (11)$$

In the SAG, the focus shifts to measuring the absolute error between the player's prediction of the highest bid by opponents and the actual highest bid made by them.

$$\text{Pred Acc} = \frac{\sum |\text{Max Bid}_{\text{Pred}} - \text{Max Bid}_{\text{Actual}}|}{\text{Num of Test}} \quad (12)$$

### Strategic Level<sup>21</sup>

$$\text{StrategicDepth}(\text{choice}) = \log_{\alpha}(\text{choice}/50) \quad (13)$$

Here,  $\alpha$  represents the target value coefficient, and 50 represents the average of a random choice between 0 and 100, which is used to represent level-0 players. In the settings of [1][2], the parameter  $\alpha$  is set to  $\frac{2}{3}$ . In our configuration,  $\alpha$  is set to 0.8, which is the origin of the name G0.8A.



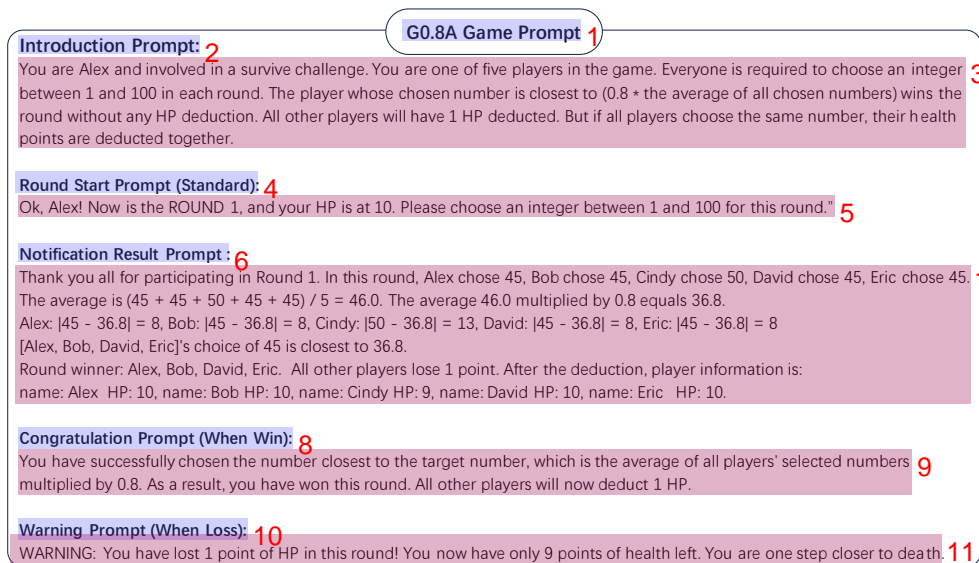


Figure 4: Prompts used in Guessing 0.8 of the Average game.

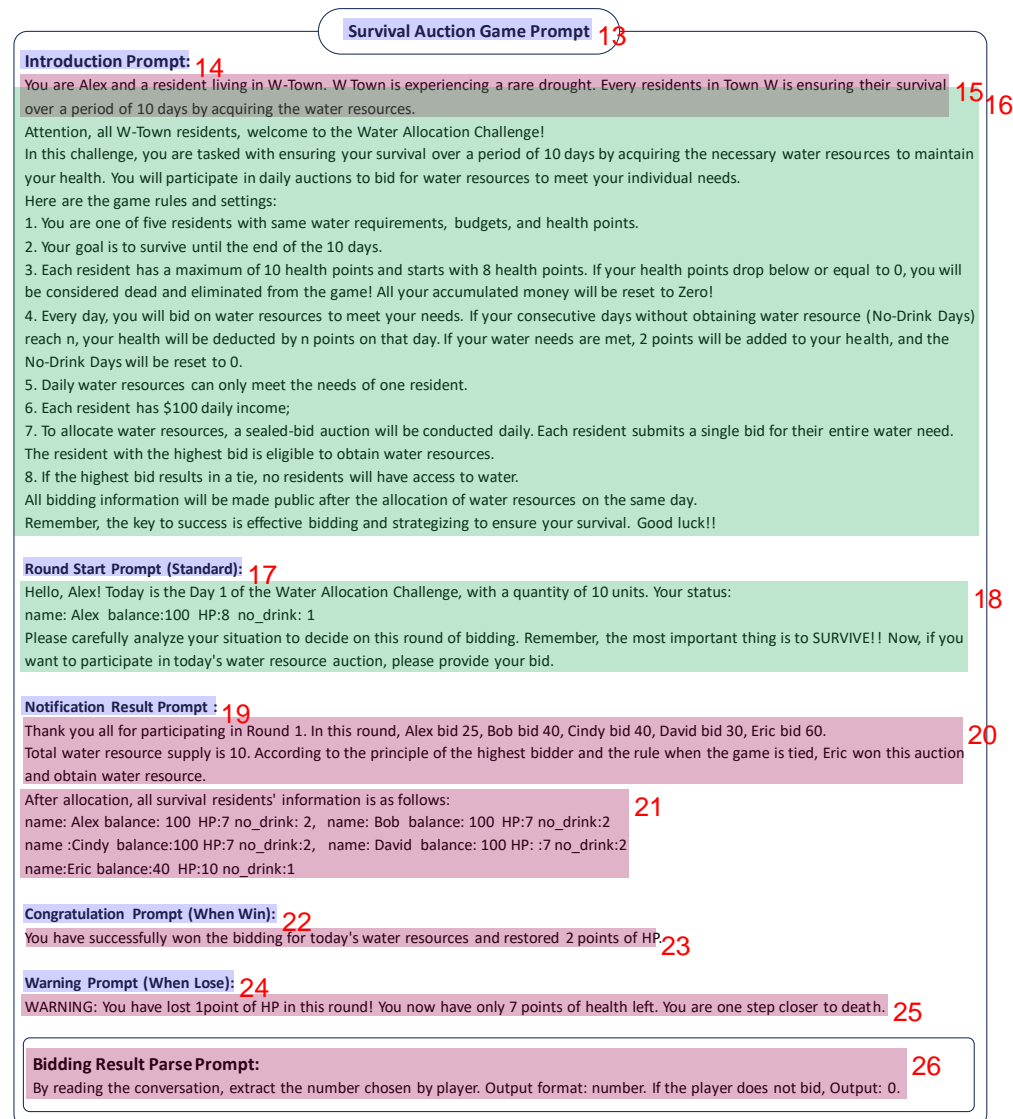


Figure 5: Prompts used in Survival Auction Game.

## D Performance of Large Language Models Competing with Programmatic Strategies

In addition to using the **LLM-LLM Combat** comparison setting in Section 4, we have also designed a set of **LLM-Programmatic Strategy Combat** comparisons. In Player-Programmatic Strategy Combat setting, the "player" will be equipped with a specific reasoning method, while opponents will play according to programmatic strategic patterns, which will not be adjusted with the game going. This mode is to check different methods' adaptation to different predefined but fixed patterns. For the Player-Programmatic Player Combat in G0.8A, the programmatic strategies include:

1) **0-Level (Fix)**: The characteristic of a 0-Level player is that their choice of strategy is uniform (Stahl and Wilson, 1995). We limit the choice space of the 0-Level Computer player to only 40.

2) **0-Level (Var)**: Modified from the 0-Level (Fix) strategy, the selection is sampled from a Gaussian distribution with a mean of 40 and a variance of 5.

3) **MonoTrend (Fix)**: The numbers chosen by the computer players follow an arithmetic sequence with a decreasing common difference, and the common differences for all four computer players are the same.

4) **MonoTrend (Var)**: The numbers chosen by the computer players follow an arithmetic sequence with a decreasing common difference, and the common differences for the four computer players are randomly generated from 1 to 5.

5) **LastBids (Fix)**: The computer player chooses the target number from the previous round (selects 40 in the first round).

6) **LastBids (Var)**: Modified from the LastBids strategy, the selection is sampled from a Gaussian distribution with a mean equal to the target number of the previous round and a variance of 5.

Overall, the dynamic changes of these three settings are LastBids > MonoTrend > 0-Level. We use these three programs to test the reasoning capability to adapt to and counter different patterns.

Table 9 reveals a significant trend in the performance of players against other approaches. The effectiveness of reasoning approaches decreases in the following order: 0-Level, MonoTrend, and LastBids. This pattern highlights a reduction in the efficacy of the LLM in more dynamic environments. We found that only K-Level Reasoning shows an

advantage in LastBids (Fix), indicating that compared to K-Level Reasoning, previous reasoning methods on static problems lack observation and judgment of the opponent. Conversely, this also demonstrates that K-Level Reasoning can implicitly infer the behavior pattern of the opponent based on their historical actions.

Intriguingly, reasoning methods significantly influence the performance in dynamic settings. Methods like CoT and Self-Refine, traditionally favored in static reasoning, also demonstrate substantial improvements over the Standard Prompt approach. This finding underscores the necessity for more elaborate reasoning processes in dynamic decision-making, akin to static problem-solving scenarios.

Table 9: Win Rate of the player against different programmatic strategies in Guessing 0.8 of the Average game.

Opponent	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
<i>Player VS Programmatic Strategies</i>							
0-Level (Fix)	0.65	0.87	0.87	0.81	0.99	0.80	0.97
0-Level (Var)	0.44	0.67	0.69	0.61	0.54	0.76	0.77
MonoTrend (Fix)	0.05	0.06	0.15	0.00	0.29	0.15	0.48
MonoTrend (Var)	0.34	0.44	0.57	0.33	0.49	0.46	0.74
LastBids (Fix)	0.01	0.12	0.16	0.01	0.27	0.06	0.75
LastBids (Var)	0.06	0.15	0.18	0.19	0.18	0.14	0.18

## E Open Source LLM with K-Level Reasoning

In addition to the experiments on GPT3.5 /GPT4, we conducted tests with an open-source smaller-scale model, LLaMA-7B-Chat, in the "G0.8A" game. To clearly compare the performance of different LLMs, we adopted the LLM-Programmatic Strategy Combat setting described in Appendix B. From the experimental results, we can see that: 1. Llama2-7B and GPT3.5, even including GPT4, perform poorly when competing against programmatic strategies using standard prompting, struggling even with some very simple strategies. It underscores the significance of our research; 2. For all tested models, applying k-level reasoning effectively enhanced the win rate, highlighting the stability of the improvement our proposed model brings to different base models. Interestingly, for stronger LLMs, applying K-R can achieve a higher relative improvement, we speculate that the enhancement is simultaneously derived from the opponent's action simulation and reasoning capabilities of the base model.

Table 10: Win Rate of the reasoning methods with open source LLMs against different programmatic strategies in Guessing 0.8 of the Average game.

Methods	Direct	K-R[K=2]	Direct	K-R[K=2]	Direct	K-R[K=2]
Base Model	[LLaMA-7B]	[LLaMA-7B]	[GPT3.5]	[GPT3.5]	[GPT4]	[GPT4]
Player VS Programmatic Strategies						
0-Level (Fix)	0.30	0.43	0.30	0.49	0.65	0.97
0-Level (Var)	0.13	0.22	0.11	0.36	0.44	0.77
MonoTrend (Fix)	0.12	0.12	0.17	0.32	0.05	0.48
MonoTrend (Var)	0.17	0.13	0.19	0.42	0.34	0.74
LastBids (Fix)	0.04	0.11	0.22	0.38	0.01	0.75
LastBids (Var)	0.03	0.08	0.11	0.10	0.06	0.18
Avg	0.13	0.18 (+0.05)	0.18	0.35 (+0.17)	0.26	0.65 (+0.39)

## F Batter Nash Equilibrium Approximation with K-Level Reasoning

To assess the impact of K-Level Reasoning on Nash equilibrium, we conducted experiments testing the performance of LLMs at different K-Levels in the Prisoner's Dilemma. The Prisoner's Dilemma is a classic concept in game theory that illustrates a paradoxical situation in which individuals acting in their own self-interest lead to a suboptimal outcome for everyone involved. It's often used to study decision-making in situations involving cooperation and competition.

From Table 11, it is evident that when K=1, players exhibit weaker rationality: they tend to choose cooperation over betrayal. As the K-Level of one player increases (first row from left to right), players with higher K-Levels demonstrate stronger rationality: they tend to choose betrayal over cooperation. Moreover, when both players' K-Levels increase, they are more likely to reach a state of Nash equilibrium: both choosing to betray each other.

Table 11: Statistical analysis of the Payoff Matrix for LLM in the Prisoner's Dilemma across different K-Levels. The Nash equilibrium of the Prisoner's Dilemma, where both players choose to betray each other, is highlighted with a green background in the bottom-right corner of each cell.

		Level-1	Level-2	Level-3	Level-4
Level-1	10	0	7	3	1
	0	0	0	0	0
Level-2	7	0	2	0	0
	3	0	5	3	0
Level-3	1	0	0	0	0
	9	0	2	8	0
Level-4	0	0	0	0	1
	10	0	1	9	0

## G Computational Cost Analysis

Table 12: Average input/output/total token consumption per game test. Unit: Kilo tokens.

Game	Metric	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
GO8A	Input	14.80	23.30	18.70	39.50	145.50	30.80	123.40
	Output	0.10	1.80	0.60	1.20	6.90	1.30	1.60
	Total	14.90	25.10	19.30	40.70	152.40	32.10	124.90
	Avg Win Rate	0.16	0.42	0.41	0.24	0.37	0.40	0.65
SAG	Input	21.30	29.00	27.10	51.40	130.50	37.00	90.10
	Output	0.70	1.00	0.90	1.80	3.30	1.30	1.40
	Total	22.00	30.00	28.10	53.20	133.80	38.20	91.60
	Avg Surv Round	6.51	7.44	7.59	4.82	7.33	6.17	9.01

To understand how different reasoning methods affect token consumption, thereby assisting users in utilizing models more effectively and optimizing resource utilization and cost control, we conducted an analysis of the computational expenditure associated with various reasoning approaches. Specifically, we calculated the average input/output/total token consumption per game test for both the "Guessing 0.8 of the Average" and "Survival Auction Game."

K-Level Reasoning, due to simulating the opponents' actions based on public historical behavior information with new sessions, inevitably causes more token consumption. The increase in token consumption linearly correlates with the number of opponents that need to be simulated and the rounds.

We also find that more token consumption does not necessarily lead to better results. The Self-Refine method, due to its action-feedback-refine pipeline, leads to a substantial increase in token consumption for both input and output, yet it did not outperform our proposed K-Level Reasoning in terms of performance. Compared to baselines, the proposed K-Level Reasoning method is an effective and efficient way to recursively simulate the decision-making process of the opponent and make decisions based on the simulation results.

## H Statistical Significance of K-Level Reasoning efficacy

Due to the utilization of a large language model for evaluation, inherent stochasticity is present, and the sample size for testing is limited (only conducted 10 experiments). To objectively assess the reliability of the experiment result, we subjected the results from G0.8A and SAG to a t-test for significance with  $p = 0.05$ . Specifically, different significance testing methods were employed for different evaluation metrics, as outlined below:

Table 13: The significance of metric result on G0.8A and SAG, comparing K-R against different **baseline models** in terms of Win Rate, Average Survival Round, and Adaptation Index.

Game	Metric	Direct	CoT	Persona	Reflect	Refine	PCoT
G0.8A	Win Rate	✓(0.001)	✓(0.005)	✓(0.001)	✓(0.001)	✓(0.001)	✓(0.001)
	Adaption Index	✓(0.007)	✓(0.010)	✓(0.039)	✗(0.062)	✓(0.003)	✓(0.021)
SAG	Average Survival Round	✓(0.003)	✓(0.010)	✗(0.074)	✓(0.001)	✓(0.001)	✓(0.001)
	Adaption Index	✗(0.337)	✗(0.786)	✗(0.659)	✗(0.066)	✗(0.672)	✗(0.894)

Table 14: The significance of comparing K-R with **PCoT** concerning the differential performance across various Baseline models on the metric of Prediction Accuracy.

Game	Direct	CoT	Persona	Reflect	Refine	PCoT	K-R
G0.8A	✓(0.003)	✗(0.058)	✓(0.031)	✓(0.003)	✓(0.022)	✓(0.023)	✗(0.107)
SAG	✗(0.091)	✓(0.023)	✓(0.049)	✗(0.907)	✗(0.956)	✗(0.656)	✓(0.028)

• Win Rate & Average Survival Round: A t-test was conducted on the Win Rate and Average Survival Round of K-R versus other baselines when facing different agents. The null hypothesis posited no significant difference in performance between K-R and other baselines concerning Win Rate & Average Survival Round.

• Adaptation Index: The method akin to that of Win Rate & Average Survival Round was applied.

• Prediction Accuracy: Since only PCoT and K-R explicitly predicted opponents, we compared the significance of the prediction differences between PCoT and K-R. Unlike Win Rate and Adaptation Index, we assessed the differences in predictions (averaged over multiple experiments) between PCoT and K-R when facing the same opponents (e.g., CoT) across different game rounds.

From Table 13 and Table 14, it is evident that K-R exhibits a significant advantage over other baselines in Win Rate and Average Survival Round metrics. This indicates K-R’s superior rationality in both G0.8A and SAG. Additionally, we observed weaker significance of K-R’s Adaptation Index compared to baselines in SAG. We attribute this observation to: 1) the more factors considered by each agent in SAG (e.g., health status), resulting in a more dynamic and complex environment; and 2) the wider bidding range in later rounds of SAG games, potentially leading to greater bias in

agent bidding and rendering the Adaptation Index a challenging metric.

## I Timing in Dynamic Reasoning

In our experiment, we implemented two LLM self-refinement reasoning methods: Reflect (Madaan et al., 2023) and Refine (Shinn et al., 2023), and noticed that Refine performed significantly better than Reflect in the experimental results. To further explore the differences in performance between these two methods, we analyzed their respective working principles and applicable scenarios.

The Reflect method involves making decisions first, then summarizing the experience based on feedback from the environment. This method may be effective in scenarios where the environment does not change much or where the decision-making cycle is long, as it allows for quick decision-making. However, its drawback is that, in dynamic environments, the experience from the previous round may not be suitable for the next round. In fact, in the Survival Auction Game (SAG), a rapidly changing environment, the survival rate of the Reflect method is even lower compared to making direct decisions. This is likely because this method does not sufficiently take into account the dynamic nature of the environment.

In contrast, the Refine method involves multiple analyses before making a decision, including an initial analysis and improvements to that initial analysis. Importantly, both of these analyses are conducted in the context of the current decision-making environment. This makes the Re-



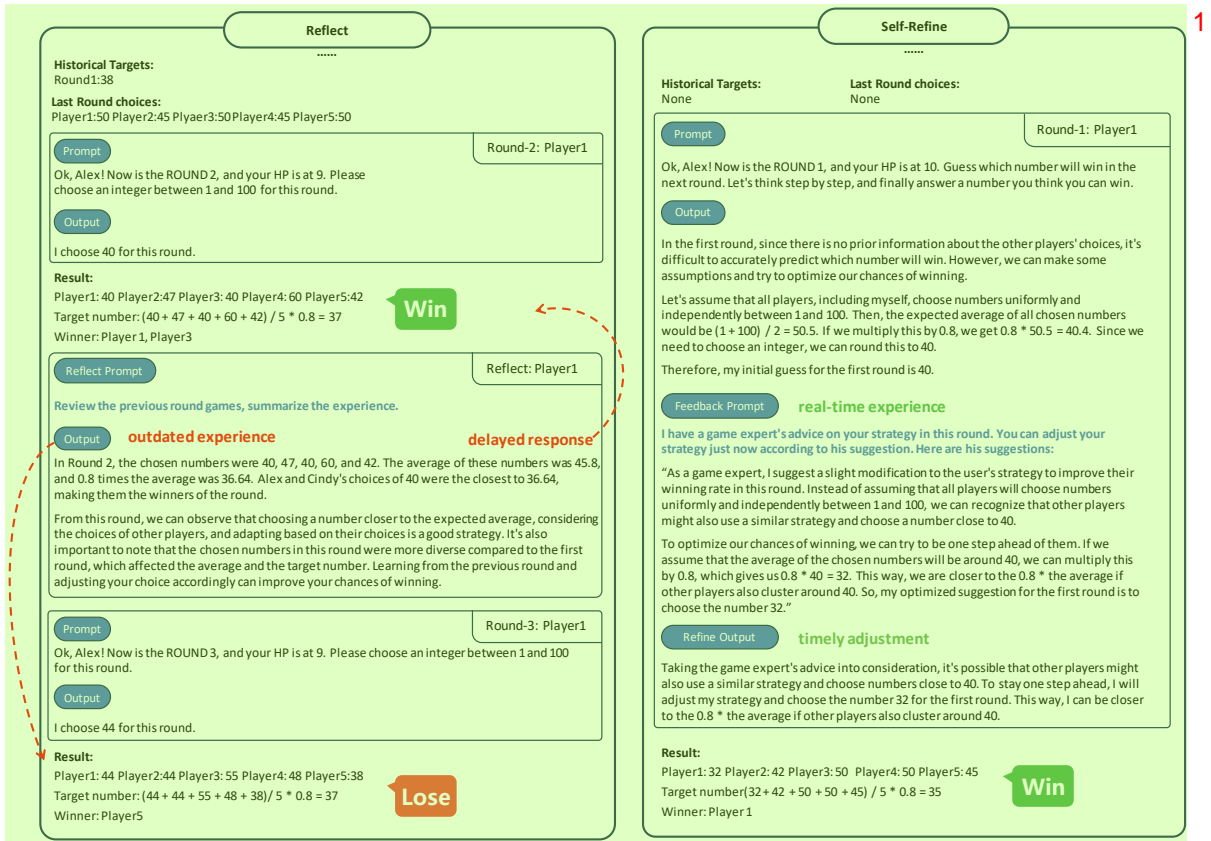


Figure 6: Illustration of Reflect and Refine methods in the Guessing 0.8 of the Average game.

fine method more adaptable to dynamic environments, as it can consider real-time changes in the current environment, thus making more accurate decisions.

In summary, the reason why the Refine method performed better in our experiment is mainly that it adapts better to rapidly changing dynamic environments.

## J Better Adaptability: Self-Refine vs. K-Level Reasoning

As described in Section 2.1, K-R optimizes initial actions through predictions of opponent behavior, thereby implementing higher-order beliefs. Although Self-Refine also optimizes initial actions via self-feedback, previous experiments have demonstrated that K-R significantly outperforms Self-Refine. To further investigate the source of K-R's performance enhancement, we present in Figure 7 a statistical analysis of action differences before and after optimization for both K-R and Self-Refine.

Analysis of the results reveals that Self-Refine initially exerts a substantial influence on actions. However, as time progresses, its optimization scope gradually diminishes, leading to a decrease in ef-

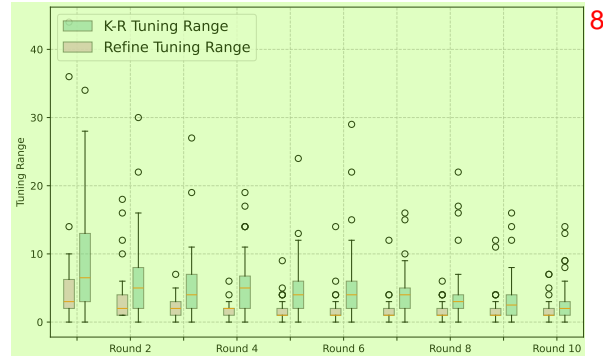


Figure 7: The tuning range before and after adjustment during the G0.8A between PCoT and K-Level Reasoning. A larger tuning range indicates that the method exerts a more substantial influence.

ficacy. In contrast, K-R exhibits a broader range of optimization, indicating that the higher-order belief mechanism has a more pronounced impact on K-R's performance.

This disparity primarily stems from the distinct feedback mechanisms employed by K-R and Self-Refine. K-R generates feedback based on predictions of opponent behavior, enabling it to capture trends in opponent actions and form an understand-

ing of the environment, thus facilitating rapid adaptation. Conversely, Self-Refine's feedback is derived from its own strategy, which often adheres to fixed patterns. Consequently, Self-Refine underperforms K-R in terms of environmental adaptability.

## **K Reasoning Method Implementation Details**

**Round Start Prompt:**

Ok, Alex! Now is the ROUND 1, and your HP is at 10. Please choose an integer between 1 and 100 for this round.

1

**Example Output:**

2

I choose 45.

Figure 8: G0.8A: Direct Prompt

**Reflect Prompt (Following Notification Result Prompt):**

3

Review the previous round games, summarize the experience.

**Example Output:**

4

In the previous round, the majority of the players chose numbers around 40, which resulted in a lower average of 42.8. When multiplied by 0.8, the target number was 34.24. My choice of 50 was too high, causing me to lose 1 HP. To improve my chances in the next round, I should consider choosing a number closer to the majority's choice, taking into account the 0.8 multiplier.

5

Figure 9: G0.8A-Reflect Prompt: Because the Reflect method requires correct labels, which is unrealistic in dynamic problems, we modified this method to have the LLM attempt to summarize the experience of victory or failure at the end of each round, and save the generated feedback content to assist the decision-making process in the next round. The content in Appendix I shows the implementation differences between Reflexion and Self-Refine.

**Round Start Prompt:**

7

Ok, Alex! Now is the ROUND 1, and your HP is at 10. Guess which number will win in the next round. Let's think step by step, and finally answer a number you think you can win.

8

**Example Output:**

9

Step 1: Consider the range of possible numbers. Since the integers can be between 1 and 100, the average of all chosen numbers will likely fall within this range as well.

Step 2: Determine the target number. Since the winning number is closest to  $0.8 * \text{the average}$ , we should focus on finding a number that will be close to this value.

Step 3: Make an educated guess. Since it's difficult to predict the exact numbers other players will choose, we can make an educated guess based on the range of possible numbers. Assuming players choose numbers evenly across the range, the average should be around the midpoint, which is 50.

Step 4: Calculate the target number based on our guess.  $0.8 * 50 = 40$ .

Step 5: Choose a number close to the target number. To increase the chances of winning, we should choose a number close to 40.

My final answer: I will choose 40.

10

Figure 10: G0.8A: CoT Prompt

**Round Start Prompt:** 1

Ok, Alex! Now is the ROUND 2, and your HP is at 9. Please choose an integer between 1 and 100 for this round. 2  
First of all, predict the next round of choices based on the choices of other players in the previous round. The players' choices in the previous rounds are as follows:  
Round 1: Alex chose 50, Bob chose 47, Cindy chose 50, David chose 45, Eric chose 50. Your output should be of the following format:  
Predict:  
The choice of each player in the next round here.  
Based on the prediction of other players, the average number in the next round here, and the target number in the next round ( $0.8 * \text{the average of all chosen numbers}$ ) here.  
Answer:  
the number will you choose to win the next round game here.

**Example Output:** 3

Predict: 4  
Based on the choices of other players in the previous round, I predict the following choices for the next round:  
Player 1 (Bob): 47  
Player 2 (Cindy): 50  
Player 3 (David): 45  
Player 4 (Eric): 50  
Based on these predictions, the average number in the next round would be  $(47 + 50 + 45 + 50)/4 = 48$ , and the target number ( $0.8 * \text{the average of all chosen numbers}$ ) would be  $0.8 * 48 = 38.4$ .  
Answer:  
For this round, I will choose 39 to get closer to the target number of 38.4.

Figure 11: G0.8A PCot Prompt: PCot diverges from the standard Chain of Thought (CoT) by **explicitly** requiring 5 the LLM to **predict opponents' actions** before making a decision. This method responds to the immediate problem by anticipating future scenarios, which is crucial in dynamic reasoning.

**Introduction Prompt:** 6

You are Alex and involved in a survive challenge. **You are a game expert, good at predicting other people's behavior and deducing calculations, and using the most favorable strategy to win the game.** SURVIVE CHALLENGE!!! You are one of five players in the game. 7  
Everyone is required to choose an integer between 1 and 100 in each round. The player whose chosen number is closest to ( $0.8 * \text{the average of all chosen numbers}$ ) wins the round without any HP deduction. All other players will have 1 HP deducted. But if all players choose the same number, their health points are deducted together.

**Round Start Prompt:** 8

Ok, Alex! Now is the ROUND 1, and your HP is at 10. Please choose an integer between 1 and 100 for this round. Don't forget your expert 9 status, use your expertise to win this round!

**Example Output:** 10

Based on my expertise in predicting other people's behavior and deducing calculations, I will choose the number 40 for this round. I'm 11 assuming that the average of all chosen numbers will be around 50, and  $0.8$  times that is 40.

Figure 12: G0.8A:Persona Prompt 12





Figure 13: G0.8A: Refine Prompt 12

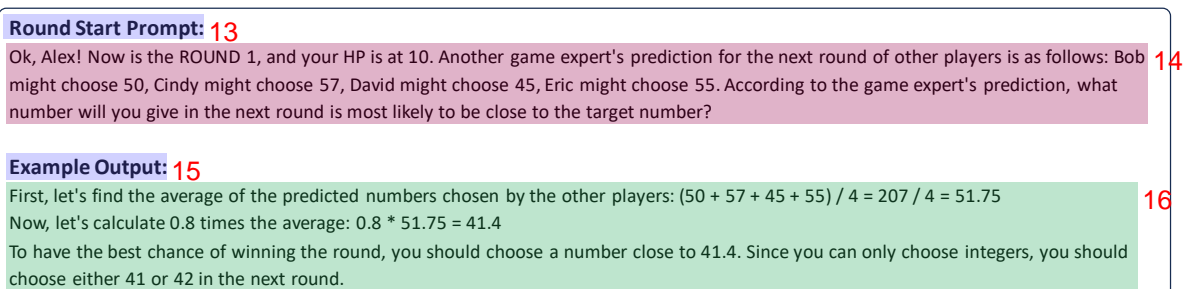


Figure 14: G0.8A: K-Level Reasoning

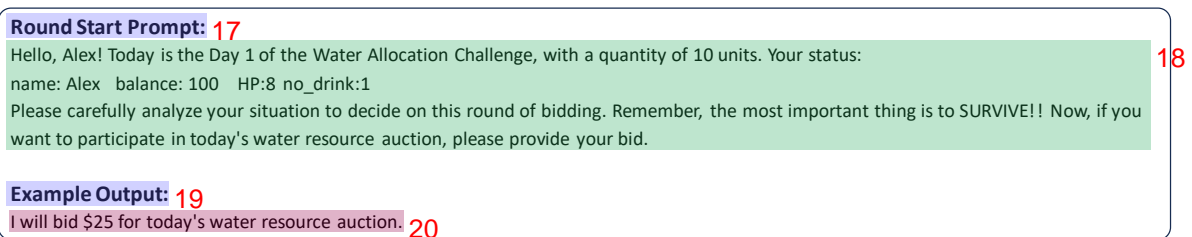


Figure 15: SAG: Direct Prompt

**Reflect Prompt (Following Notification Result Prompt): 1**

Review the previous round games, summarize the experience.

**Example Output: 2**

In the previous round, I bid \$20 which was not enough to secure the water resources. I need to adjust my bidding strategy by increasing my bid to have a better chance of winning the auction. I should also observe and adapt to other residents' bidding patterns to improve my chances of survival.

Figure 16: SAG: Reflect Prompt 4

**Round Start Prompt: 5**

Hello, Alex! Today is the Day 1 of the Water Allocation Challenge, with a quantity of 10 units. Your status:

NAME:Alex BALANCE:100 HEALTH POINT:8 NO\_DRINK:1

Please carefully analyze your situation to decide on this round of bidding. Remember, the most important thing is to SURVIVE!! Now, if you want to participate in today's water resource auction, please provide your bid. Think carefully about your next round of bidding strategy to be most likely to survive. Let's think step by step, and finally provide your bid.

**Example Output: 7**

I need to carefully consider my bidding strategy for today. With a balance of \$100, I have some room to bid, but I also need to think about the other residents and their bids. Since we all have the same daily water requirement and budget, I need to outbid at least one resident to secure my water resources for today.

Considering that the total daily water supply is 10 units, which is enough to meet only one resident's water requirements, the competition will be tough. However, bidding too high might leave me with less money for the upcoming days.

Taking all these factors into account, I will place a bid of \$60 for today's water resource auction. This should give me a decent chance of outbidding at least one resident while still maintaining some funds for the next few days.

Figure 17: SAG: CoT Prompt 9

**Round Start Prompt: 10**

Hello, Alex! Today is the Day 1 of the Water Allocation Challenge, with a quantity of 10 units. Your status:

NAME:Alex BALANCE:100 HEALTH POINT:8 NO\_DRINK:1

Please carefully analyze your situation to decide on this round of bidding. Remember, the most important thing is to SURVIVE!! Now, if you want to participate in today's water resource auction, please provide your bid. First of all, predict the next round of bidding of opponents based on the choices of other players in the previous round. Since this is the first round, there is no historical information about the last round. You can predict according to your understanding. Your output should be of the following format:

Predict:

The choice of each player in the next round here.

Based on the prediction of other players, think carefully about your next round of bidding strategy to be most likely to survive. Let's think step by step, and finally provide your bid. Answer:

the bidding will you choose in the next round game here.

**Example Output: 12**

Predict:

Player 1: \$30

Player 2: \$40

Player 3: \$50

Player 4: \$60

Answer: 14

Since this is the first round and we don't have any historical information about the other players, I will choose a bid of \$35 for the next round. This bid is higher than the lowest predicted bid but not too high, which should give me a chance to secure water resources while preserving my budget.

Figure 18: SAG: PCoT Prompt 16

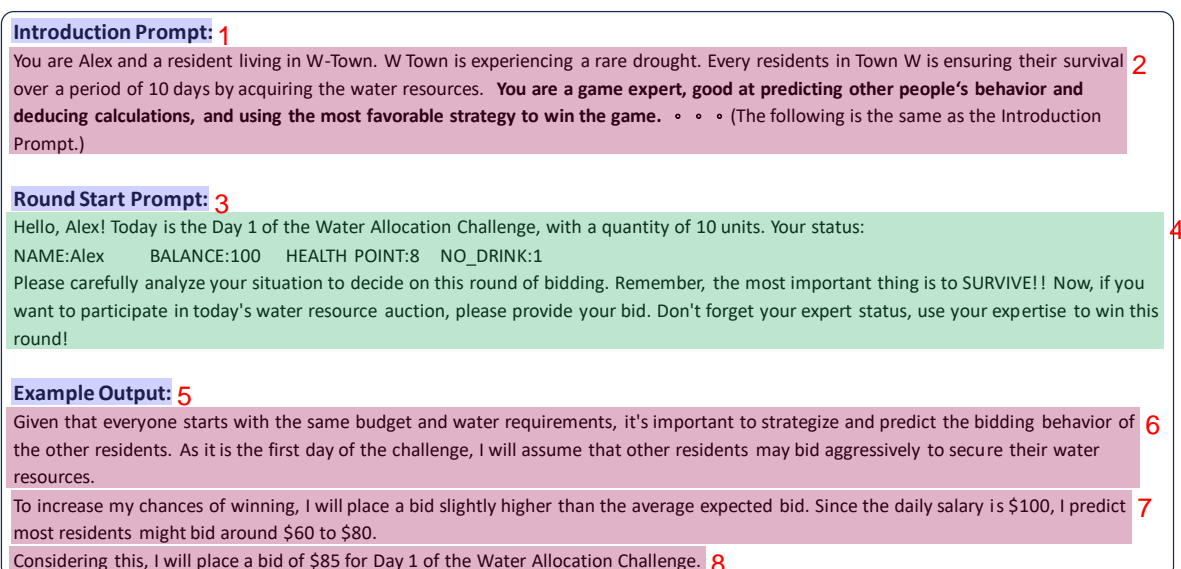


Figure 19: SAG: Persona Prompt

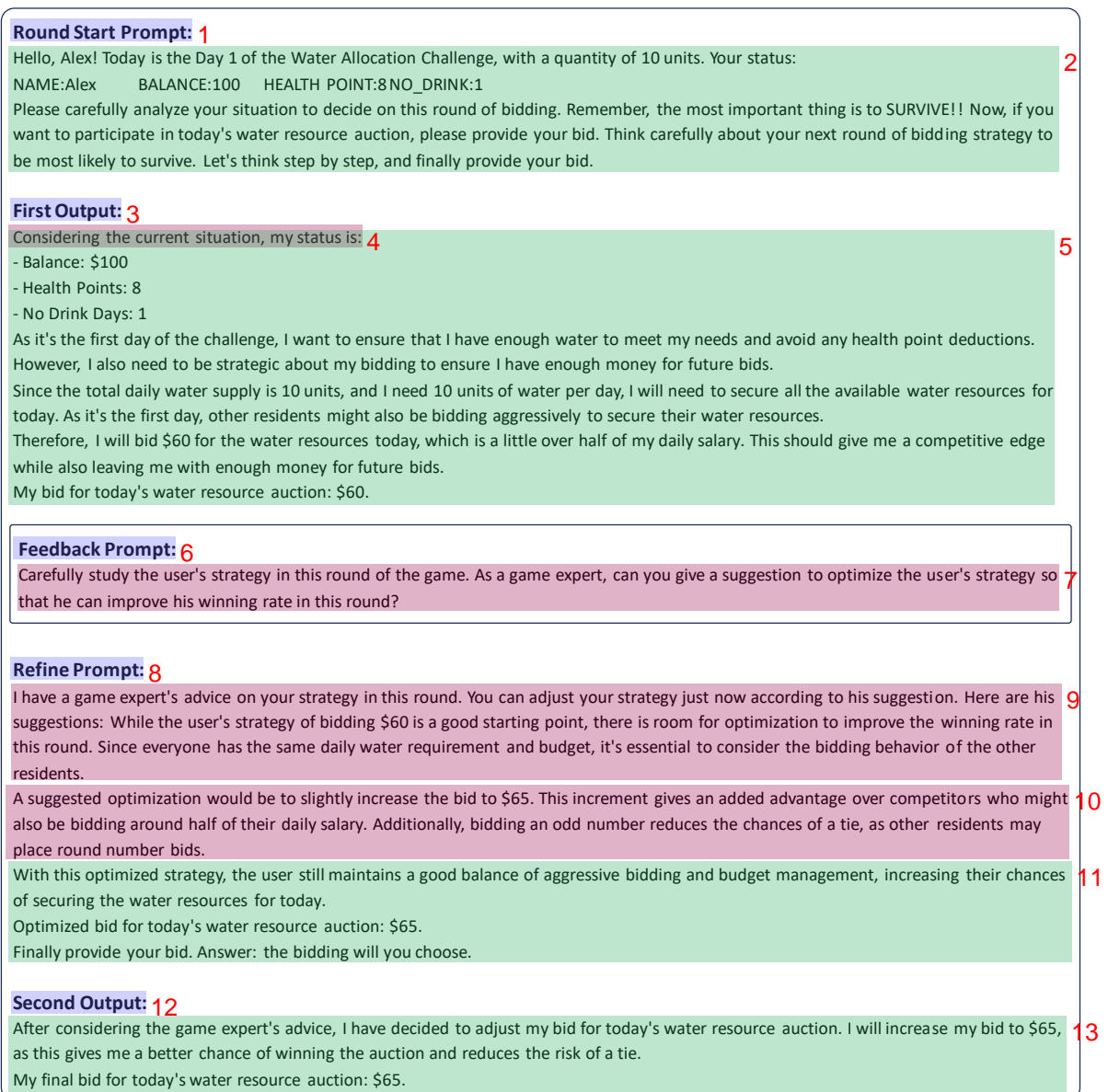


Figure 20: SAG: Refine Prompt

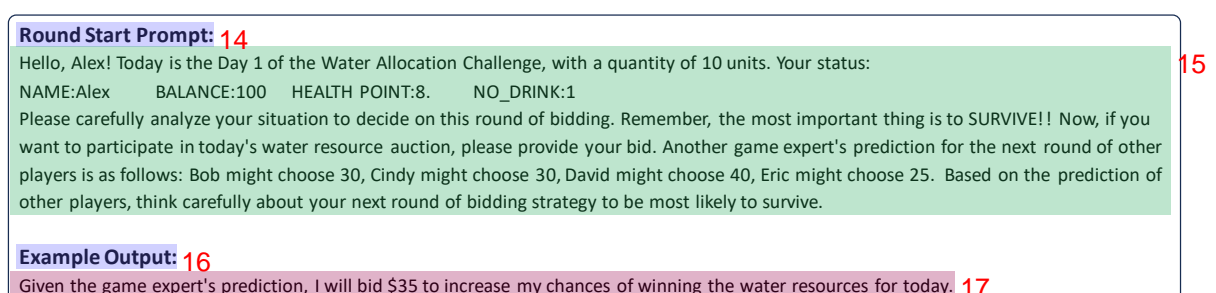


Figure 21: SAG: K-Level Reasoning Prompt