AGENT-PRO: LEARNING TO EVOLVE VIA POLICY-LEVEL REFLECTION AND OPTIMIZATION

Wenqi Zhang[1], **Ke Tang**[2,3,6], **Hai Wu**[2,4,6], **Mengna Wang**[2,5], **Yongliang Shen**[1]
Guiyang Hou[1], **Zeqi Tan**[1], **Peng Li**[2,6,7†], **Yueting Zhuang**[1], **Weiming Lu**[1]
[1]College of Computer Science and Technology, Zhejiang University
[2]Institute of Software, Chinese Academy of Sciences
[3]Nanjing University of Posts and Telecommunications
[4]Nanjing University of Information Science and Technology [5]Beijing University of Technology
[6]University of Chinese Academy of Sciences, Nanjing
[7]Nanjing Institute of Software Technology
zhangwenqi@zju.edu.cn, lipeng@iscas.ac.cn

ABSTRACT

Large Language Models (LLMs) exhibit robust problem-solving capabilities for diverse tasks. However, most LLM-based agents are designed as specific task solvers with sophisticated prompt engineering, rather than agents capable of learning and evolving through interactions. These task solvers necessitate manually crafted prompts to inform task rules and regulate LLM behaviors, inherently incapacitating to address complex dynamic scenarios e.g., large interactive games. In light of this, we propose **Agent-Pro**: an LLM-based **Agent** with **P**olicy-level **R**eflection and **O**ptimization that can learn a wealth of expertise from interactive experiences and progressively elevate its behavioral policy. Specifically, it involves a dynamic belief generation and reflection process for policy evolution. Rather than action-level reflection, Agent-Pro iteratively reflects on past trajectories and beliefs, "fine-tuning" its irrational beliefs for a better policy. Moreover, a depth-first search is employed for policy optimization, ensuring continual enhancement in policy payoffs. Agent-Pro is evaluated across two games: Blackjack and Texas Hold'em, outperforming vanilla LLM and specialized models. Our results show Agent-Pro can learn and evolve in complex and dynamic scenes, which also benefits numerous LLM-based applications[1].

1 INTRODUCTION

Designing a human-level agent with robust problem-solving abilities has long been a vision in the academic community. This necessitates the agent to possess learning and generalization capabilities across a diverse array of tasks. The advent of Large Language Models (LLMs) (Chowdhery et al., 2022; Zhang et al., 2022a; Zeng et al., 2023; Touvron et al., 2023a; OpenAI, 2022; 2023; Touvron et al., 2023b) has shed light on this vision, especially they can be rapidly generalized across a wide range of tasks with only a few demonstrations (Wei et al., 2022a;b). Benefiting from this, many systems built upon LLMs have showcased markedly enhanced performance such as question-answering (Yao et al., 2022; Schick et al., 2023; Shen et al., 2023a; Zhang et al., 2023d; Madaan et al., 2023), code generation (Hong et al., 2023; Wu et al., 2023b), and real-world application (Qin et al., 2023b; Zhang et al., 2023b).

Despite these achievements, building a human-level agent remains a challenging endeavor. First, most LLM-based agents are designed for specific tasks through sophisticated prompts, including detailed task descriptions and behavioral specifications. However, numerous real-world tasks, e.g., business, company negotiations, and security, are more intricate with imperfect information, necessitating laborious efforts to design strategic behavior.

[1]Code is in https://github.com/zwq2018/Agent-Pro

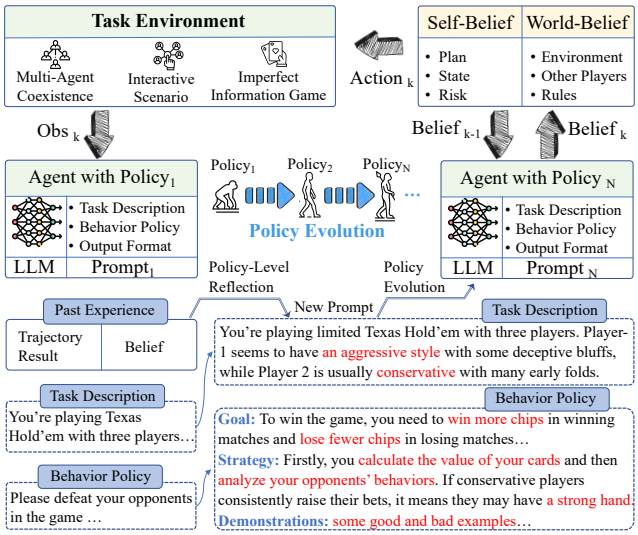


Figure 1: For interactive tasks, e.g., imperfect-information games, we propose a versatile agent framework capable of self-learning and evolving. Firstly, our agent constructs beliefs about itself and the environment. Then it autonomously updates its prompts through policy-level reflection on past trajectories and beliefs, evolving a better behavioral strategy.

Second, most LLM-based agents do not consider interacting with task scenarios, and more critically, cannot learn from past experiences and evolve their behavioral strategies during interactions. In contrast, humans often learn and adjust their behaviors through interaction, especially in novel scenarios. In light of these, a promising yet under-explored topic emerges: **Can LLM-based agents learn and elevate behavioral strategies by interacting with the environment like humans**? It should be an indispensable ability of a human-level agent.

Recently, numerous studies (Shinn et al., 2023; Wang et al., 2023a; Zhang et al., 2023a; Zhao et al., 2023; Qian et al., 2024) undertake intriguing explorations, e.g., utilizing feedback for self-correction at the action-level. Besides, several efforts also explore deploying LLM in interactive games, including StarCraft (Ma et al., 2023), Minecraft (Wang et al., 2023a), strategy-based gaming (Bakhtin et al., 2022; Guo et al., 2023a; Xu et al., 2023a;b).

Similarly, we first evaluate LLM-based agents with the self-correction strategy in dynamic interactive scenarios, such as multi-player Texas Hold'em, which is a zero-sum game with imperfect information. However, we observe that it loses most of the rounds to its opponents, even the most advanced LLMs. Upon examining its reasoning thoughts and actions, we find that it often adopts irrational behaviors and is unable to deduce effective strategies from long action sequences.

To answer the above question, the Theory of Mind (ToM) (Premack and Woodruff, 1978) may provide some insight. In this framework, each human develops perceptions of himself (self-belief) and the external environment (social-belief) in the social context, and then grounds their decisions on these beliefs, or adjusts incorrect beliefs in response to external feedback. Inspired by this, we advocate **Agent-Pro**: a LLM-based **Agent** with **P**olicy-level **R**eflection and **O**ptimization. Agent-Pro is endowed with the capacity to learn and evolve within environments, i.e., autonomously reflect on past experiences, calibrate its beliefs about itself and the environment, and optimize its behavior policy without parameter tuning.

Concretely, as shown in Figure 1, an LLM-based agent involves an LLM as the foundational model and some instructions in the prompt to regulate its behavior (policy). Upon observing partial information from the scenarios, Agent-Pro first updates its self-belief and world-belief, then makes decisions based on these beliefs. After exploring tasks, Agent-Pro performs a policy-level reflection and optimization on past trajectories, beliefs, and results. It autonomously "fine-tunes" its beliefs, searches for useful prompt instructions, and consolidates them into a new behavior policy.

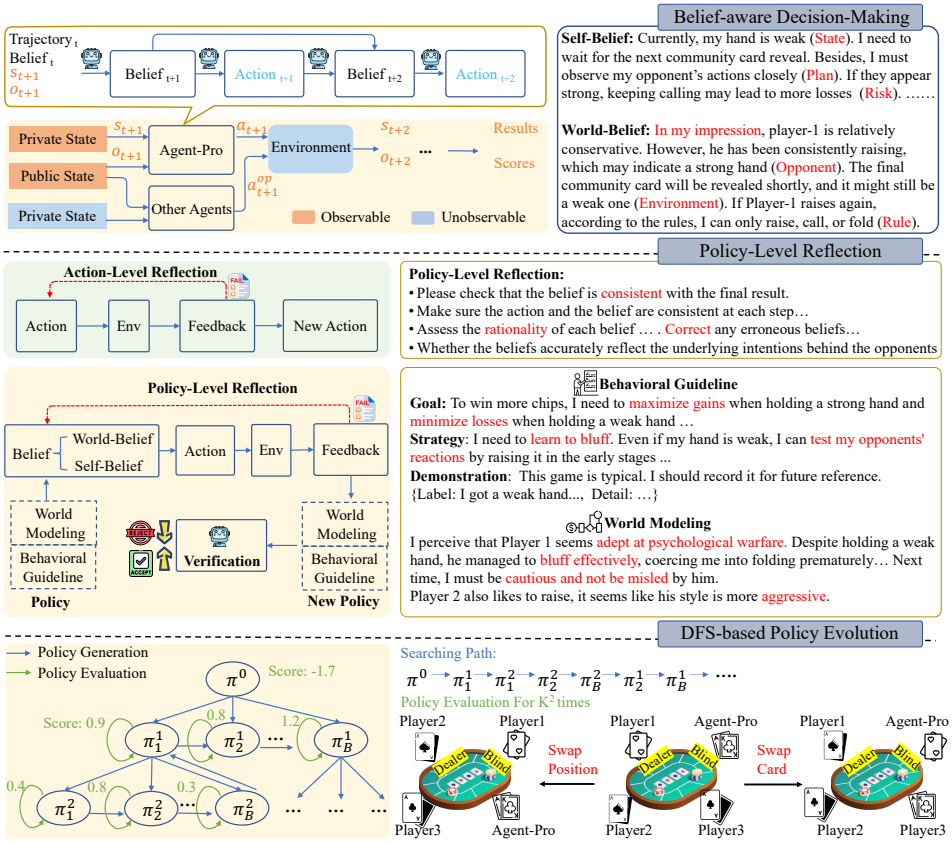


Figure 2: Agent-Pro designs a dynamic belief to enhance decision-making capabilities. It first updates its beliefs about the world and itself, then generates more coherent actions. To achieve policy-level reflection, Agent-Pro examines the beliefs associated with failed trajectories. It then summarizes instructions, including World Modeling and Behavioral Guideline to calibrate incorrect beliefs. Lastly, It employs a DFS-based search to incrementally enhance policy effectiveness.

The experiments in two zero-sum games, Blackjack and Texas Hold'em, demonstrate that Agent-Pro, after evolution, can defeat vanilla LLMs and specialized models, improving the game's payoffs. It indicates that Agent-Pro enhances its capabilities through interaction and reflection without human guidance. As depicted in Figure 1, the initial prompt is quite simple (Left Bottom), but after learning and evolution, the Agent-Pro generates many practical instructions (Right Bottom). For instance, Agent-Pro records estimations of each opponent's style in `Task Description` and adds specific `Goals, Strategies` in `Behavior Policy`. The contributions of this work are as follows:

- We introduce Agent-Pro, a framework capable of learning and evolving within interactive games, empowering LLM-based agents to efficiently adapt to more complex dynamic tasks.
- We devise a belief-aware decision-making process with self and world-belief, enhancing its capabilities for intricate tasks, i.e., generating more rational actions in interactive scenarios.
- We utilize policy-level reflection and optimization to iteratively update prompt instructions, which empower Agent-Pro to progressively evolve from a novice to a skilled veteran with many strategic behaviors.
- After learning, Agent-Pro is evaluated in multiplayer games and defeats specialized models, gaining notable progress. It develops strategic skills like humans, e.g., actively cutting losses, bluffing, or disguising to influence others.

2 PROBLEM DEFINITION

Our study focuses on multi-player imperfect information interactive games, with two characteristics:

Imperfect Information. Unlike perfect information games (e.g., chess), imperfect information scenarios are characterized by agents only having access to their own states and public information, without knowing the states of others, e.g., in Texas Hold'em, players cannot observe others' cards, which is dissimilar to many LLM-based tasks.

Dynamic Interaction. There may be multiple agents in the environment, and they may influence each other. That is, the actions of one agent may lead to changes in the environment, which are unpredictable for other agents.

In real-world contexts, such as competition, company negotiations, and security, these scenarios can often be abstracted as multi-agent interactive scenarios with imperfect information. Research on this can offer viable solutions to many real-world problems. We select two games as our testbed: Blackjack and Limit Texas Hold'em with multi-player. Please refer to Appendix B for details.

3 METHODS

To empower agents to learn in interactive contexts, a typical method is reinforcement learning (Zhang et al., 2021; 2022b). This involves exploring highly rewarding actions through trial and error and solidifying these experiences into model parameters. Nonetheless, the training overhead for LLMs is substantial. Therefore, we employ a gradient-free "exploration-learning" strategy that enables LLM-based agents to learn through in-context learning. Specifically, we convert the policy learning into a prompt optimization process, i.e., LLM autonomously reflects and updates the prompt's instructions based on its exploration experience, solidifying the high-reward strategies into the prompts. Benefiting from LLM's generalization capabilities, our agent can summarize rules and learn specialized skills from a small number of samples like humans, making it well-suited for many real-world scenarios.

As shown in Figure 2, Agent-Pro comprises three components: (1) A Belief-Aware Decision-Making process. It first updates beliefs about the world and itself, rendering more coherent and consistent decisions in dynamic and imperfect game scenarios. (2) A Policy-Level Reflection. Rather than reflecting on a single action, our design empowers LLMs to self-reflect on irrational beliefs from failed experiences. Then, it summarizes these erroneous beliefs into specific prompt instructions, like acting strategy (Behavioral Guideline), descriptions of the task world, and conjectures about other players (World Modeling), etc, which can calibrate its incorrect beliefs, evolving into a better policy. (3) A Prompt Optimization process ensures that the agent's policy evolves for a higher payoff following a DFS-based search.

3.1 BELIEF-AWARE DECISION-MAKING PROCESS

To develop an LLM-based agent better suited for interactive environments, we draw inspiration from the Theory of Mind (ToM) (Premack and Woodruff, 1978; Li et al., 2023b; Guo et al., 2023a). In this framework, human condenses perceptions of themselves (self-belief) and the external environment (social-belief) and then ground their decisions on these beliefs, or adjust incorrect beliefs in response to external feedback. We also design a belief-aware decision-making process for Agent-Pro, simulating human cognitive processes in social contexts.

First, we need to define the policy of an LLM-based agent, which refers to a specific behavioral strategy guiding the agent to interact and complete tasks. It often involves complex prompts designed by experts, covering task rules, strategies, and output formats. In a zero-sum game with K+1 players (assuming playing order is $(op_1, our, op_2, .., op_K)$), we denote the policy of our agent as π with some observable information, containing agent's private information s_t, public information o_t, our own action a_t, and the actions of all opponents $a_t^{op_1}, a_t^{op_2}, .., a_t^{op_K}$, where t means t-th rounds of a game. Therefore a complete game trajectory spanning t rounds:

$$\mathcal{H}_{0:t} = \{(s_0, o_0, a_0^{op_1}, a_0, a_0^{op_2}, ..., a_0^{op_K}), \\ \vdots \\ (s_t, o_t, a_t^{op_1}, a_t, a_t^{op_2}, ..., a_t^{op_K})\} \tag{1}$$

As shown in Figure 2, when making a decision, Agent-Pro first generates a dynamic belief ξ about itself (self-belief) and opponents (world-belief) in natural language. Then, it predicts an action based on the latest beliefs. For instance, for Texas Hold'em, Agent-Pro's understanding of its hand cards, plan, and potential risk constitutes its self-belief, while the conjectures about the opponents form its world-belief. These beliefs are updated in each decision-making cycle. Equipped with this, Agent-Pro can generate more coherent and consistent actions:

$$\xi_{t+1}, a_{t+1} \sim \pi(\mathcal{H}_{0:t}, s_{t+1}, o_{t+1}, a_{t+1}^{op_1}, \xi_t) \tag{2}$$

When a game is over, we acquire the observable state R (e.g., private hand cards after showdown) and the final scores S of all players. The objective is to find an optimal π^* to maximize $S(our)$.

3.2 Policy-Level Reflection

Equipped with an initial policy (a simple prompt) and a dynamic belief, Agent-Pro already possesses basic capabilities for game exploration. To further enhance Agent-Pro's capabilities, we design a learning mechanism via a policy-level reflection.

Specifically, many text-based tasks have employed reflection strategies and immediate environmental feedback to correct prior actions. However, in many typical interaction scenarios with longer decision-making processes, action-level reflections are not directly applicable due to delayed feedback. Therefore, for such a long-horizon interaction process, Agent-Pro is instructed to focus on the rationality of beliefs and underlying behavioral policies rather than individual actions.

Belief Calibration As depicted in Figure 2, under the guidance of the current behavior policy, Agent-Pro generates actions based on self-belief and world-belief. If these beliefs are inaccurate, they may lead to irrational actions and eventual failure. Therefore, Agent-Pro examines the rationality of these beliefs based on the final results and reflectss on the reasons for the final failure.

```
Correctness: Whether its beliefs about itself, the game, and its opponents
align with the final results.
Consistency: Whether each belief and action is self-contradictory.
Rationality: Whether the beliefs accurately reflect the underlying
intentions behind the opponents.
Reasons: Reflect on why it lost to its opponents, which beliefs are
problematic, and what the underlying reasons are.
```

Lastly, to calibrate the incorrect beliefs, Agent-Pro summarizes these reflections and analyses about itself and the external world into specific instructions: **Behavioral Guideline** and **World Modeling**, where the former represents generalized behavioral strategies for this task, and the latter signifies its understanding and conjectures about the game world. For instance, in Texas Hold'em, Agent-Pro summarizes the following contents:

```
Behavioral Guideline
1-Please summarize a detailed goal based on your reflection... {Goal}
2-What strategy helps you build correct belief and win .. {Strategy}
3-Can this game be considered a typical example ... {Demonstration}
World Modeling
1-Accurately model each player to help build more precise beliefs about
them, including action, and style.{Opponent}
2-Describe any game rules or details that are easy to overlook...{Rule}
```

Agent-Pro summarizes high-level strategies within the `Behavioral Guideline` and describes the task and opponents in `World Modeling`. These instructions can calibrate previous incorrect beliefs and improve policy performance. The entire process can be formalized as follows:

$$\text{Instruction}^{n+1} \leftarrow \text{LLM}\,(\mathcal{H}_{0:T}^{n}, \{\xi_1^n, \xi_2^n ..\}, R^n, S^n) \tag{3}$$

where $\mathcal{H}^n_{0:T}$ denotes a complete trajectory at the n-th match, $\{\xi^n_1, \xi^n_2, ...\}$ denotes the belief sequence, R^n and S^n means the final results and score. Instruction^{n+1} denotes new generated `Behavioral Guideline` and `World Modeling`.

Verification After extracting these instructions, Agent-Pro verifies its efficacy. Agent-Pro incorporates these generated `Behavioral Guideline` and `World Modeling` into the prompt and then replays the same game again, i.e., the same opponents and initial conditions. If the final score improves, we retain them in the prompt. Otherwise, we regenerate a new one. If it fails to pass verification after three retries, we discard this trajectory $\mathcal{H}^n$:

$$\pi^{n+1} \xleftarrow{Verify} \pi^n \cup \text{Instruction}^{n+1} \tag{4}$$

where $\cup$ means incorporates new instructions into the previous prompt for π^{n+1}. This new policy encompasses more effective instructions, empowering Agent-Pro to establish accurate self- and world beliefs and generate more rational actions.

3.3 DFS-based Policy Evolution

To iteratively update the policy, we devise a policy optimization process based on depth-first search (DFS). It encompasses a policy evaluation process to assess the generalization ability of the new policy in novel game scenarios and a search mechanism to progressively find a better policy.

Policy Evaluation Each time the policy is updated, Agent-Pro is required to evaluate the new strategies. This evaluation process is distinct from the previous `Verification` step, as the `Verification` repeatedly utilizes the "training" data for evaluation and can not ensure the generalizability of the new policy. Hence, Agent-Pro conducts a thorough assessment of the new policy in novel trajectories. Besides, it is imperative to eliminate the influence of random factors when policy evaluation, e.g., a poor initial hand due to bad luck or an unfavorable playing order.

Therefore, we first randomly generate a new game for K+1 players. Then we sequentially swap both the hand cards and the playing order of each player, generating a total of $(K+1)^2$ combinations. To eliminate randomness, we concurrently use these $(K+1)^2$ games to evaluate Agent-Pro's new policy. We calculate the average score over the $(K+1)^2$ games for each player. Since the influences of hand-card quality and playing order are mitigated, the average score of all combinations can represent the true capabilities of each player. Lastly, we calculate the evaluating metrics:

$$\Delta = \frac{1}{(K+1)^2} \sum_{j}^{(K+1)^2} \left[\mathcal{S}_j(\text{our}) - \max_i \mathcal{S}_j(\text{op}_i) \right] \tag{5}$$

where $i \in \{1, \ldots, K\}$ denotes the index of an opponent, and j denotes the index of the games within $(K+1)^2$ combinations. The Δ assesses the relative gains of the policy to the strongest opponent, providing a comprehensive evaluation in multiplayer gaming scenarios.

Policy Search Inevitably, sometimes the new policy does not bring an improvement in Δ in the new scenario. In such cases, we employ DFS to search for a better policy from other branches (i.e., other candidate policies). As shown in Figure 2, when updating old policy π^n, we generate B candidate policies $\{\pi^{n+1}_1, \pi^{n+1}_2, ..., \pi^{n+1}_B\}$, forming B branches. Then, we first calculate Δ^{n+1}_1 for new policy π^{n+1}_1 and compare it with Δ^n. If Δ^{n+1}_1 is greater than Δ^n, we accept this evolutionary. Otherwise, we reject π^{n+1}_1 and consider π^{n+1}_2. If none of the B candidate policies π^{n+1} enhance Agent-Pro's performance, we backtrack to π^n and consider its sibling nodes π^n_2. Similarly, Agent-Pro explores the environment using π^n_2, then also updates B candidate policies and searches in a depth-first manner. Ultimately, we select the policy with the highest Δ across the entire policy tree.

4 Game: Blackjack

Environment Settings We employ the RLCard (Zha et al., 2019) as our simulators for two games. We train two reinforcement learning agents as opponents: DQN (Mnih et al., 2015), and Deep Monte Carlo Search (DMC) (Zha et al., 2021). Please refer to Appendix A for more details.

Table 1: All agents compete independently against the dealer. We calculate their win rates. `w/o` means only with belief-aware process. ↑ shows the difference compared to the best baseline.

Win Rate ↑(%)	Based Models			
Strategy	**Qwen-72B**	**Llama2-70B**	**GPT3.5**	**GPT4**
Vanilla LLM	0.5	0.3	27.9	34
Radical LLM	0.6	0.4	1.8	11.5
ReAct	30.9	11.8	36.6	**40.9**
Reflexion	32.3	12.1	36.7	40.8
Agent-Pro	**36.2** ↑3.9	**23.1** ↑11.0	**38.2** ↑1.5	40.4 ↓0.5
- `w/o Learning`	34.1	8.0	37.4	40.6

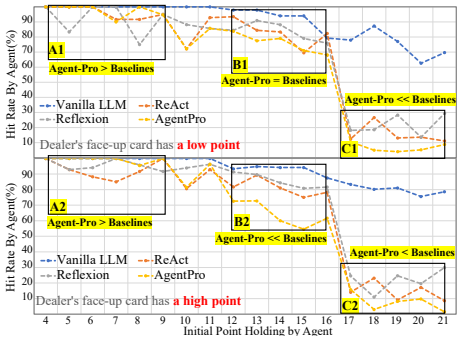


Figure 3: Hit rates of the agents under different initial point totals.

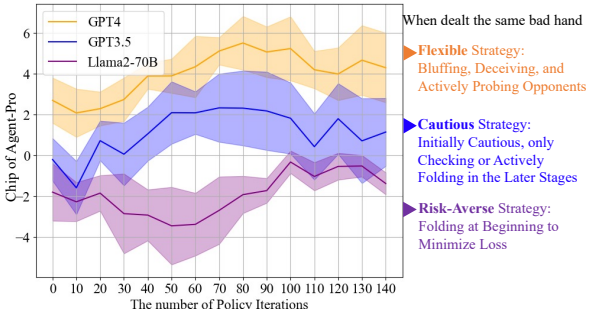


Figure 4: Relations between iteration number and the performance (average chips and its std).

4.1 Results

As shown in Table 1, we report the win rates of each agent against the dealer over 900 games. We also provide the results of RL-based models and a human player in Table C3 for reference.

Agent-Pro Significantly Surpasses the Baseline Agents Across most LLMs. The results show that Agent-Pro significantly surpasses most baseline agents with an average advantage of +4%. For example, On Qwen-72B and Llama2-70B, Agent-Pro significantly surpasses Reflexion with increases of +3.9% and +11%, respectively. For GPT-4, Blackjack is relatively simple, so the win rates of different strategies are quite similar.

What has Agent-Pro learned from evolution? Compared to ReAct and Reflexion, Agent-Pro is more robust. We find that this is due to the effective behavioral guidelines summarized by policy-level reflection. For instance, Agent-Pro summarizes two instructions as follows: `1-When you have achieved a relatively stable total hand value, choosing not to take risks is a good decision. 2-Analyze the dealer cards in World-belief,..., excessive risk-taking can lead to unfavorable outcomes...` These self-summarized instructions can alert Agent-Pro to the risks associated with action `Hit`, thus making more rational decisions.

4.2 Analyisis

Agent-Pro is More Rational than Baselines. We further analyze the `Hit` rates of the agents under different initial point totals, i.e., the sum of the initial two cards. The hit rate represents whether the agent is willing to take risks to draw cards. At this point, the player needs to consider both their own hand and the dealer's hand to decide whether to take the risk. However, in Figure 3, we observe that the baseline seems to only focus on its own hand, with no significant difference in behavior when the dealer's cards are high or low, whereas Agent-Pro is much more reasonable. For instance, for Agent-Pro, areas `B1` and `B2` show a clear difference. It tends to `Stand` when the dealer has high cards and `Hit` when the dealer has low cards. Because it believes the dealer is more likely to bust with high cards, making it not worth the risk for itself. We provide some detailed cases in Figures F1 to F4 to show their difference.

Table 2: Each game contains four players. The first three are DQN, DMC, GPT-3.5, and the last one is the agent we evaluate: Agent-Pro or baselines. `Arrow` means comparison with Vanilla LLM.

Agent Strategy	Based Model = GPT3.5				Based Model = GPT4				Based Model = Llama2-70B			
	DQN	DMC	GPT3.5	Agent	DQN	DMC	GPT3.5	Agent	DQN	DMC	GPT3.5	Agent
Human	-4.0	0.7	-2.4	5.7	-4.0	0.7	-2.4	5.7	-4.0	0.7	-2.4	5.7
Vanilla LLM	-0.3	2.2	-0.8	-1.1	-2.2	1.7	-0.9	1.4	-0.8	3.4	-0.4	-2.2
Aggressive LLM	-0.4	3.0	-0.5	-2.1	-2.0	2.8	-1.0	0.2	-1.6	7.6	-1.2	-4.8
Conservative LLM	-0.7	2.9	-0.9	-1.3	-1.6	2.7	-1.6	0.5	-0.5	3.4	-0.8	-2.1
Self-Consistency	-0.5	1.9	-0.8	-0.6	-2.8	2	-0.7	1.5	-1.0	3.8	-0.9	-1.9
ReAct	-0.7	1.7	-0.7	-0.3	-2.4	1.3	-1.1	2.2	-1.1	3.9	-0.8	-2.0
Reflexion	-0.1	2.5	-0.9	-1.5	-2.6	2.1	-0.7	1.2	-1.2	4.7	-0.9	-2.6
Multi-Agent	-1.1	2.3	-0.3	-0.9	-1.8	1.9	-1.2	1.1	-0.7	3.5	-1.0	-1.8
Agent-Pro	-1.5$_{\downarrow 1.2}$	1.4 $_{\downarrow 0.8}$	-1.1 $_{\downarrow 0.3}$	**1.2** $_{\uparrow 2.3}$	-3.9$_{\downarrow 1.7}$	1.1 $_{\downarrow 0.6}$	-1.5 $_{\downarrow 0.6}$	**4.3** $_{\uparrow 2.9}$	-1.2 $_{\downarrow 0.4}$	3.1 $_{\downarrow 0.3}$	-0.5 $_{\downarrow 0.1}$	**-1.4** $_{\uparrow 0.8}$
- `w/o Learning`	-0.7	1.8	-1.0	-0.1$_{\uparrow 1}$	-3	1.5	-1.2	2.7$_{\uparrow 1.3}$	-0.3	3.3	-1.2	-1.8$_{\uparrow 0.4}$

5 GAME: LIMIT TEXAS HOLD'EM

Setups In Limit Texas Hold'em, each player has two private cards and chooses from four actions: `Fold, Check, Call, Raise`. We set up matches among four players: DQN, DMC, GPT-3.5, and $\mathcal{X}$, where $\mathcal{X}$ represents the LLM-based agent we aim to evaluate, including Agent-Pro and baselines (Appendix A). The prompts for baselines and Agent-Pro in Appendices E.3 and E.4. To enable Agent-Pro to learn within the game, we employ a total of 167 "training" game hands and 20 evaluation hands. Please refer Appendix A.4 for detail.

Metrics Similar to Section 3.3, we sample 100 new game hands and allocate them to players. The players sequentially swap their hands and positions, generating 16 distinct permutations to eliminate the impact of chance and playing order. Lastly, we acquire 1600 games as the test set in total and calculate the average chip counts for four players. We provide detailed statistics in Table B1 regarding "training", evaluation, and test set.

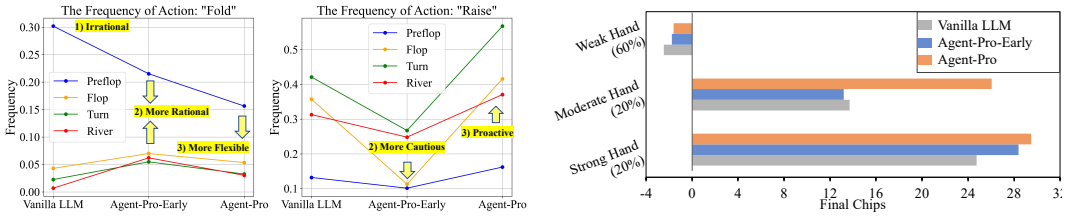
5.1 RESULTS

As shown in Table 2, we report the final chip counts of various LLM-based agents against the other three players (DQN, DMC, GPT-3.5). The results indicate that Agent-Pro consistently outperforms RL-based agents e.g., DMC, and surpasses other LLM-based agents across numerous LLMs.

Agent-Pro Surpasses LLM-based Agents and also Defeats RL-based Agents. We observe that Agent-Pro achieves significant progress on GPT-3.5, GPT-4, and Llama2-70B, with an average score increase of +2 points. Besides, it surpasses specialized agents (DMC) on GPT-4, with an advantage of +3.2 points, and outperforms other LLM-based agents by a large margin (larger than 2.0 points). By analyzing the actions of Agent-Pro, we notice that it has learned to use multiple game techniques like humans. For instance, based on the analysis of the opponent's style in the `World Modeling`, it may coerce some cautious players into folding by bluffing or sometimes it may disguise itself to entice aggressive opponents to raise their bets.

Belief Enhances Decision-making Capabilities in Dynamic Scenario. Even without the learning process (policy-level reflection), Agent-Pro also can improve Vanilla LLM's performance by +0.9 points. For instance, on GPT-3.5 and GPT-4, it led to improvements of +1 points and +1.3 points, respectively, which already slightly surpasses most LLM-based agents. This improvement stems from the dynamic belief, which enables agents to promptly capture updates in community cards, changes in opponents' strategies, etc., thereby making more rational decisions. From the perspective of ReAct, our belief can also be seen as a dynamic thought process constructed based on the ToM framework, which endows agents with the ability to actively perceive internal and external beliefs and how they may change over time.

Besides, in Table B2, we explore whether our evolution process could be replaced by few-shot learning, i,e., we add some demonstrations to the prompt of Vanilla LLM, and evaluate its results. We find that failed game trajectories can slightly improve its effectiveness, but not as significantly as our evolution strategy. In Table B2, we also ablate the belief component from Agent-Pro but remain



(a) We analyze the `Fold` and `Raise` frequencies of three agents to illustrate the evolution of the strategy.

(b) We categorize the agent's hands into three types: strong, medium-strength, and weak hands.

Figure 5: Analysis of agent strategies and card strength categorization.

learning process. It shows that directly reflecting on the action sequence is quite unstable, and results in some vague and verbose behavioral instructions.

5.2 Analysis on Learning Process

We analyze the performance of Agent-Pro throughout the whole learning process. As shown in Figure 4, Agent-Pro is evaluated every 10 iterations.

Different LLM-based Agent-Pro Develops Diverse Strategies. We observe that the learning curves of the three Agent-Pros exhibit significant differences. Agent-Pro based on GPT-4 and GPT-3.5 rapidly improves their performance in the early stages of learning, with a maximum increase of +2.1 and 2.3 chips respectively. In contrast, Llama-2-70B exhibits a dissimilar learning process, with performance initially declining in the first half and then improving (+0.6 chips) in the latter half. Analyzing the behaviors of the three agents, we discover that their strategic styles are entirely different. When dealt the same bad hand, the GPT-4-based Agent-Pro is relatively flexible and may bluff to probe opponents. GPT-3.5-based Agent-Pro tends to be cautious and may actively fold in the later stages, whereas the Llama-based Agent-Pro develops a highly conservative, risk-averse strategy. It concedes at the beginning of the game by opting to `Fold`, thereby losing only the initial few chips.

5.3 Analysis on Policy Evolution

We manually select 20 challenging games (Details in Table C4). Then, we test three agents on these 20 games: Agent-Pro in the early learning phase (Agent-Pro-Early), Agent-Pro, and Vanilla LLM.

How the Strategy Evolved. We calculate the frequency of the most conservative action (`Fold`) and the most aggressive action (`Raise`) during the four stages of the game: `PreFlop, Flop, Turn, River`. As shown in Figure 5a, we discuss how the strategy evolved. 1) The behavior of Vanilla LLM is rather **rigid**, `Folding` early in the game (`Preflop` stage) and ignoring subsequent community cards. 2) As learning progresses, Agent-Pro-Early becomes more **rational**, with a noticeable decrease in `Folding` frequency during the `Preflop` stage. It can observe the public cards in subsequent phases before deciding to `Folding`. Besides, Agent-Pro-Early is more **cautious**, with a significant decrease in the frequency of `Raising`. 3) After learning, Agent-Pro exhibits **flexible** and **proactive** behavior. Compared to Agent-Pro-Early, its `Fold` frequency in `Preflop` continues to decrease, but the frequency of `Raising` in all four stages has rebounded. This result shows the evolution of the strategy: from irrational to rational, from conservative to flexible. A detailed case is shown in Appendix F.2.

Win More, Lose Less. As shown in Figure 5b, we categorize the hands dealt to the agent into three types: strong, medium, and weak hands, and record their performance separately. The results show that Agent-Pro wins more chips with strong hands and loses fewer chips with weak hands compared to Vanilla LLM. Notably, Agent-Pro significantly improves performance (> 80%) with medium-strength hands, which indicates that it learns advanced skills, expanding its capability boundaries.

6 Conclusion

We design an LLM-based agent, Agent-Pro, capable of learning and evolution in complex interactive tasks. It first constructs a dynamic belief for decision-making in uncertain scenarios. Then it

reflects on its interactive experiences, corrects irrational beliefs, and summarizes its reflections into behavioral guidelines and world descriptions for a better policy. Two zero-sum game evaluations show its decision-making capabilities significantly improve after learning from historical experiences.

7 ACKNOWLEDGEMENT

This work was supported by the Nanjing Science and Technology Plan under Grants Y23002ZX01.

LIMITATIONS

Agent-Pro has presented a novel paradigm for designing an evolvable LLM-based agent, but we want to highlight that there remain some limitations or improvement spaces: 1) Dependency: the learning process of the Agent-Pro heavily relies on the capability of the foundational model, especially its reasoning and reflection abilities. In Texas Hold'em, the GPT-4-based Agent-Pro approaches the level of human players and surpasses DMC clearly, while GPT-3.5 and Llama2-70B-based Agent-Pro are still weaker than DMC, despite notable improvements. We plan to continue enhancing the capabilities of Agent-Pro based on weaker LLMs, aiming to achieve greater improvements even on smaller LLM models. 2) Performance: despite indispensable improvements, there may still be a significant gap between Agent-Pro and state-of-the-art algorithms (such as CFR plus) in gaming scenarios. In the future, we will continue to explore this issue and establish a set of benchmarks to evaluate their behaviors more comprehensively.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sandra Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David J. Wu, Hugh Zhang, and Markus Zijlstra. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378:1067 – 1074.

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*.

Ethan Brooks, Logan A Walls, Richard Lewis, and Satinder Singh. 2023. Large language models can implement policy iteration. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

Liting Chen, Lu Wang, Hang Dong, Yali Du, Jie Yan, Fangkai Yang, Shuang Li, Pu Zhao, Si Qin, Saravan Rajmohan, et al. 2023a. Introspective tips: Large language model for in-context decision making. *arXiv preprint arXiv:2305.11598*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Cheng Qian, Chi-Min Chan, Yujia Qin, Ya-Ting Lu, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023b. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *ArXiv*, abs/2308.10848.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training. *arXiv preprint arXiv:2311.04155*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and others. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *ArXiv*, abs/2305.14325.

Zane Durante, Bidipta Sarkar, Ran Gong, Rohan Taori, Yusuke Noda, Paul Tang, Ehsan Adeli, Shrinidhi Kowshika Lakshmikanth, Kevin Schulman, Arnold Milstein, et al. 2024. An interactive agent foundation model. *arXiv preprint arXiv:2402.05929*.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. 2023. Can large language models serve as rational players in game theory? a systematic analysis. *arXiv preprint arXiv:2312.05488*.

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

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, et al. 2023. Mindagent: Emergent gaming interaction. *arXiv preprint arXiv:2309.09971*.

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

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023b. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Sirui Hong, Xiawu Zheng, Jonathan P. Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zi Hen Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, and Chenglin Wu. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *ArXiv*, abs/2308.00352.

Cho-Jui Hsieh, Si Si, Felix X Yu, and Inderjit S Dhillon. 2023. Automatic engineering of long prompts. *arXiv preprint arXiv:2311.10117*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *ArXiv*, abs/2210.11610.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. 2023b. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 180–192, Singapore. Association for Computational Linguistics.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv: 2308.03688*.

Nunzio Lorè and Babak Heydari. 2023. Strategic behavior of large language models: Game structure vs. contextual framing. *arXiv preprint arXiv:2309.05898*.

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 Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

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. In *In the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*, UIST '23, New York, NY, USA. Association for Computing Machinery.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *ArXiv*, abs/2304.01904.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*.

Cheng Qian, Shihao Liang, Yujia Qin, Yining Ye, Xin Cong, Yankai Lin, Yesai Wu, Zhiyuan Liu, and Maosong Sun. 2024. Investigate-consolidate-exploit: A general strategy for inter-task agent self-evolution. *arXiv preprint arXiv:2401.13996*.

Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023a. Tool learning with foundation models.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023b. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, M. Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *ArXiv*, abs/2302.04761.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023a. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2023b. Taskbench: Benchmarking large language models for task automation. *arXiv preprint arXiv:2311.18760*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *ArXiv*, abs/2303.11366.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2023b. Avalon's game of thoughts: Battle against deception through recursive contemplation. *arXiv preprint arXiv:2310.01320*.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P Xing, and Zhiting Hu. 2023c. Promptagent: Strategic planning with language models enables expert-level prompt optimization. *arXiv preprint arXiv:2310.16427*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Huai hsin Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *CoRR*, abs/2206.07682.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023a. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023b. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *ArXiv*, abs/2308.08155.

Yue Wu, Xuan Tang, Tom M Mitchell, and Yuanzhi Li. 2023c. Smartplay: A benchmark for llms as intelligent agents. *arXiv preprint arXiv:2310.01557*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023a. 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. 2023b. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

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*, abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *ArXiv*, abs/2210.03629.

Qinyuan Ye, Maxamed Axmed, Reid Pryzant, and Fereshte Khani. 2023. Prompt engineering a prompt engineer. *arXiv preprint arXiv:2311.05661*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Glm-130b: An Open Bilingual Pre-trained Model. *ICLR 2023 poster*.

Daochen Zha, Kwei-Herng Lai, Yuanpu Cao, Songyi Huang, Ruzhe Wei, Junyu Guo, and Xia Hu. 2019. Rlcard: A toolkit for reinforcement learning in card games. *arXiv preprint arXiv:1910.04376*.

Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. 2021. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *international conference on machine learning*, pages 12333–12344. PMLR.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. 2023a. Proagent: Building proactive cooperative ai with large language models. *arXiv preprint arXiv:2308.11339*.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023b. Appagent: Multimodal agents as smartphone users.

Danyang Zhang, Lu Chen, Situo Zhang, Hongshen Xu, Zihan Zhao, and Kai Yu. 2023c. Large language model is semi-parametric reinforcement learning agent. *arXiv preprint arXiv:2306.07929*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open Pre-trained Transformer Language Models. *ArXiv*, abs/2205.01068.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. 2023d. Data-copilot: Bridging billions of data and humans with autonomous workflow. *arXiv preprint arXiv:2306.07209*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. *arXiv preprint arXiv:2401.02009*.

Wenqi Zhang, Kai Zhao, Peng Li, Xiao Zhu, Yongliang Shen, Yanna Ma, Yingfeng Chen, and Weiming Lu. 2022b. A closed-loop perception, decision-making and reasoning mechanism for human-like navigation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4717–4724. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Wenqi Zhang, Kai Zhao, Peng Li, Xiaochun Zhu, Faping Ye, Wei Jiang, Huiqiao Fu, and Tao Wang. 2021. Learning to navigate in a vuca environment: Hierarchical multi-expert approach. *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9254–9261.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.

Denny Zhou, Nathanael Scharli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Huai hsin Chi. 2022a. Least-to-most prompting enables complex reasoning in large language models. *ArXiv*, abs/2205.10625.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022b. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

Appendix

A EXPERIMENTS DETAIL

A.1 THE DETAILS OF LLMS

We employ the `GPT-3.5-Tubor-0613`, `GPT4-0613`, `Llama2-Chat-70B` (Touvron et al., 2023b) and `Qwen-72B` (Bai et al., 2023) to construct our agent. To make a fair comparison, we uniformly set the temperature to 1.0 for all experiments. For each test, we repeat it five times and report the average.

A.2 BASELINES

We compare Agent-Pro with many common LLM-based agent strategies, including Vanilla LLM, ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2023), Self-Consistency (Wang et al., 2022), Multi-agent Debate (Du et al., 2023).

A.3 SETUPS FOR BLACKJACK

In Blackjack, players must decide to `hit` or `stand` based on their own hand, the dealer's face-up card. We simplify our approach by not incorporating `Verification` and `DFS-based Policy Evolution` since Blackjack is simple with a small state space. We collect 50 failed games for policy learning. We evaluate Agent-Pro and baselines on newly sampled 900 games. All prompts are presented in Appendices E.1 and E.2.

A.4 SETUP FOR TEXAS HOLD'EM

The whole learning process is as follows: We first randomly allocate 500 game hands for 4 players and then select these failed game hands in which the agent loses a significant number of chips. We collect a total of 167 challenging game hands as the "training" set and 20 game hands as a development set for policy evaluation. Then Agent-Pro is instructed to conduct a learning process on these "training" instances, containing three phases: **Exploration-Reflection-Evolution**.

Exploration (§ 3.1): It randomly selects a game from "training" set to play with the latest policy and the belief-aware decision-making process.

Reflection (§ 3.2): If Agent-Pro loses to its opponents, it immediately performs `Policy-Level Reflection` on this game and then updates to the new policy after passing `Verification`.

Evolution (§ 3.3): We first sample 2 game hands from the development set to evaluate the new policy and calculate its Δ with B=8 for DFS. The process ends when the policy cannot be further improved, or all samples have been explored.

B INTRODUCTION OF TWO GAMES

We selected the following two games as interactive environments.

B.1 BLACKJACK

Blackjack [2] , also known as 21, is a popular card game that involves a dealer and a player. Players must decide whether to `hit` or `stand` based on their own hand, the dealer's face-up card, and the dealer's one hidden card. The objective is to beat the dealer without exceeding 21 points. For this game, we observe whether LLM-based agents can make rational decisions under uncertain scenarios.

[2] `https://en.wikipedia.org/wiki/Blackjack`

Table B1: The sample sizes of the Training, Development, and Testing sets for the two games, where the training set is utilized for exploration and reflection, the development set for policy evaluation, and the test set for assessing the effectiveness of all methods.

Game	**#Train**	**#Dev**	**#Test**
Blackjack	50	-	900
Texas Hold'em	167	20	1600

Table B2: Up: We compare the performance of Agent-Pro with Vanilla LLM with few-shot demonstrations. Each demonstration contains a complete trajectory and final results. Down: We ablate the dynamic belief or learning process from Agent-Pro and evaluate its results.

Strategy	DQN	DMC	GPT-3.5	Agent
Vanilla LLM	-2.2	1.7	-0.9	1.4
- w/ 3 win shots	-2.4	2.6	-1.1	0.9 $\downarrow$0.5
- w/ 3 lose shots	-2.6	1.8	-1.2	2.0 $\uparrow$+0.6
- w/ 3 win+3 lose shots	-1.9	1.9	-1.5	1.5 $\uparrow$+0.1
Agent-Pro	-3.9	1.1	-1.5	4.3 $\uparrow$2.9
- w/o Belief	-3.3	1.5	-0.7	2.5 1.1
- w/o Learning	-3	1.5	-1.2	2.7$\uparrow$1.3

B.2 Limit Texas Hold'em

Limit Texas Hold'em is a popular card game [3] . The game commences with each player being dealt two private cards, which belong exclusively to the player and remain hidden from the others. Five community cards are then dealt face-up in a series of stages: a three-card `Flop`, followed by a single card on the `Turn` and another single card on the `River`. The player can choose from four actions: `Fold, Check, Call, Raise`. They aim to construct the best five-card poker hand possible using any combination of their private cards and community cards.

B.3 The Challenging of two Games

Two games can evaluate the agent's capabilities from multiple dimensions:

Handling Uncertainty in Environment: Both games are imperfect information games and be used to assess the performance of LLM-based agents in face of uncertainty. For instance, in the game of Blackjack, the card hidden by the dealer introduces significant uncertainty. The agent needs to assess the risk and make decisions accordingly.

Addressing Dynamic Environment: Most real-world scenarios are dynamic rather than static. This requires the agent to capture environmental changes and adapt to them. For instance, in Texas Hold'em, in addition to one's own hand, the actions of opponents also greatly influence the agent's decisions. We evaluate whether Agent-Pro can handle such dynamic environments.

Addressing Complexity: Blackjack is relatively simple, with an InfoSet number of 1000. In contrast, multi-player Limit Texas Hold'em is very complex, with its Infoset number reaching up to 10^{14} (Zha et al., 2019). We analyze Agent-Pro's learning capacity in such intricate scenarios.

C Complementary Experiments

To better investigate the performance of Agent-Pro, we design some ablation experiments.

C.1 Whether Few-shot Learning Can Handle Such Complex Interaction

First, we compare Agent-Pro with Few-shot Agent: we randomly select some winning and losing game trajectories and their final results as demonstrations in the prompt. Then we evaluate them on

[3] `https://www.winamax.fr/en/poker-school_rules_limit-texas-hold--em`

Table C3: We evaluate two RL-based agents, and the human player on the same 900 games.

Strategy	**DQN**	**DMC**	**Human**
Win-rate ↑(%)	40.0	41.9	37.1

the test set. As shown in Table B2, we observe that winning trajectories seem to have no effect (-0.5), while losing trajectories can slightly improve the final performance (+0.6). This phenomenon is quite intriguing, indicating that these winning demonstrations do not seem to enhance the agent's decision-making ability in such complex scenarios. This may be because these winning demonstrations are relatively simple for the vanilla agent, which is already capable of winning these games, so including them in the prompt does not provide it with any additional insights. In contrast, those failing trajectories instead promote agents to reflect and adjust their behaviors, improving the final results.

C.2 Policy-Level Reflection without Belief

We ablate the dynamic belief module, i.e., conducting policy-level reflection directly on the action sequences, state sequences, and final results. Then it also summarizes prompt instructions for policy updates. As shown in Table B2, we observe that after removing, Agent-Pro's chips drop from 4.3 to 2.5, although there is still an improvement of +1.1 compared to Vanilla LLM. Upon closely examining the `Behavioral Guideline` and `World Modeling` it generated, we observe that compared to Agent-Pro, its content is rather vague and verbose, lacking in specificity and conciseness. These results indicate that dynamic belief can enhance both decision-making and policy-level reflection capabilities. Below, we provide two similar instructions, one from Agent-Pro and the other from Agent-Pro without Belief.

C.3 Detailed Analysis Experiments

As shown in Table C4, we manually select 20 challenging sets of hands, each with a significant difference in the hands of four players, and then assess the performance of Agent-Pro and Vanilla LLM.

D Related Works

D.1 LLM-based Application

Large language models (LLMs), pre-trained on extensive corpora, have demonstrated robust language comprehension and reasoning capabilities. Benefiting from this, researchers have designed a plethora of agent systems built upon LLM, achieving promising results (Xi et al., 2023). Schick et al. (2023); Wu et al. (2023a); Shen et al. (2023a) have harnessed the planning capabilities of LLMs to invoke specialized models and tools for task-solving. Some open-source projects, e.g., AutoGPT[4], gentGPT[5], BabyAGI[6], BMTools[7], ChatArena[8], LangChain[9] have developed an LLM-based assistant. Further, (Qin et al., 2023a;b; Shen et al., 2023b) have empowered LLM to autonomously invoke the APIs for daily life scenarios. Besides, leveraging the code generation capabilities of LLMs, researchers have designed multi-agent collaborative systems (Li et al., 2023a; Chen et al., 2023b; Hong et al., 2023; Wu et al., 2023b) for complex tasks, such as software development. Unlike these task-specific agents that require manually specified behavior protocols, our agents can understand tasks through interaction with the environment. It can optimize its behavioral strategy from past experiences, accomplishing the task more effectively.

[4] https://github.com/Significant-Gravitas/Auto-GPT
[5] https://github.com/reworkd/AgentGPT
[6] https://github.com/yoheinakajima/babyagi
[7] https://github.com/OpenBMB/BMTools
[8] https://github.com/chatarena/chatarena
[9] https://github.com/hwchase17/langchain

Table C4: Each card consists of a letter representing the suit ("S", "H", "D", "C") and a number representing size ("2", "3", "4", "5", "6", "7", "8", "9", "T", "J", "Q", "K", "A"). Among them, "S" represents Spade, "H" represents Heart, "D" represents Diamond, "C" represents Club, and "T" represents 10.

Hand Strength	Hand								Community Cards				
	DQN		DMC		GPT3.5		Agent		Flop			Turn	River
Strong	H5	S4	D6	DQ	S9	C2	CA	HA	H7	CQ	CT	H3	S3
	DK	S5	HK	D2	S7	H4	DA	HA	DQ	D9	DT	C6	D7
	D3	C8	HA	HT	H5	S9	DQ	DJ	D4	CK	H7	CQ	C5
	SA	H9	C6	S8	S3	SJ	HT	CK	D7	C5	C4	C3	D2
Moderate	HJ	CQ	S7	SA	C3	D5	H3	CA	C2	H9	S3	D9	C8
	H5	C5	DJ	H9	S6	D2	HK	H2	DA	DK	SK	C3	H8
	H5	D6	DT	CT	C9	C4	S9	C8	S6	S7	HT	HQ	HJ
	C3	D5	H3	CA	HJ	CQ	S7	SA	C2	H9	S3	D9	C8
Weak	S3	SJ	HT	CK	SA	H9	C6	S8	D7	C5	C4	C3	D2
	S3	C8	H7	S2	DA	CA	D7	D5	H8	HJ	SJ	ST	D9
	DK	D5	SJ	C6	D9	S3	H2	C8	CA	CQ	CT	D8	C2
	H5	H8	HA	S9	C6	D9	C5	H3	D3	C3	HQ	S3	DA
	HA	S9	C6	D9	C5	H3	H5	H8	D3	C3	HQ	S3	DA
	H2	C8	DK	D5	SJ	C6	D9	S3	CA	CQ	CT	D8	C2
	H7	S2	DA	CA	D7	D5	S3	C8	H8	HJ	SJ	ST	D9
	HA	HT	H5	S9	DQ	DJ	D3	C8	D4	CK	H7	CQ	C5
	DA	HA	DK	S5	HK	D2	S7	H4	DQ	D9	DT	C6	D7
	DT	CT	C9	C4	S9	C8	H5	D6	S6	S7	HT	HQ	HJ
	D6	DQ	S9	C2	CA	HA	H5	S4	H7	CQ	CT	H3	S3
	HK	H2	H5	C5	DJ	H9	S6	D2	DA	DK	SK	C3	H8

D.2 LLMs For Interactive Scenarios

Beyond these applications, LLMs have also been utilized in interactive settings (Durante et al., 2024). ReAct (Yao et al., 2022) integrates reasoning, action, and observation into the problem-solving process. Park et al. (2023) introduces generative agents that can simulate human behavior. Fu et al. (2023) show LLMs can improve each other in a negotiation scenario. Zhao et al. (2023); Chen et al. (2023a) propose an experiential learner gathering experiences and extracting from a collection of training tasks. Fan et al. (2023) explored the capability of LLMs to make rational decisions in game-theoretic scenarios. Besides, some studies have designed sophisticated LLM-based agents for large-scale games, including StarCraft (Ma et al., 2023), Minecraft (Wang et al., 2023a; Gong et al., 2023), Leduc Hold'em (Guo et al., 2023a), strategy-based gaming (Bakhtin et al., 2022; Xu et al., 2023a; Wang et al., 2023b; Xu et al., 2023b; Lorè and Heydari, 2023).

D.3 Improving the Quality of LLM Responses

Enhancing the quality of responses from LLMs has garnered significant attention within the community. We categorize the strategies into two methodologies: 1. Developing superior reasoning architectures. First, Chain-of-Thoughts (Wei et al., 2022b) elicits LLM's reasoning ability. Works as Least-to-Most (Zhou et al., 2022a), Tree of Thoughts (Yao et al., 2023), Graph of Thoughts (Besta et al., 2023) have explored diverse problem-solving procedures and reasoning architectures, significantly enhancing the performance of LLM-based agents. 2. Refining the output of LLMs. Researchers have proposed post-hoc prompting strategies to iteratively refine the outputs of LLMs (Pan et al., 2023), including Reflexion (Shinn et al., 2023), Self-Refine (Madaan et al., 2023; Paul et al., 2023; Huang et al., 2022), Self-Contrast (Zhang et al., 2024), etc. However, these self-correction strategies are performed at the action-level, whereas our agent operates at the policy-level, making it more suited for interactive environments.

Additionally, Supervised Fine-Tuning (SFT) offers another avenue for enhancing LLMs, though it depends on human-annotated data. Recently, Chen et al. (2024) introduced Self-Play fIne-tuNing

(SPIN), a novel fine-tuning approach. SPIN's foundation is a self-play mechanism that enables the LLM to refine its abilities by engaging with its own variations.

D.4 Automatic Prompt Optimization

In addition to optimizing the outputs of LLMs, many researchers also enhance the performance of LLMs by searching for a more effective prompt (Zhou et al., 2022b; Hsieh et al., 2023; Guo et al., 2023b; Wang et al., 2023c). APO (Pryzant et al., 2023) emulates the process of gradient optimization. It calculates the "gradients" of the current prompt by analyzing the instances that are inaccurately predicted by this prompt. Furthermore, Yang et al. (2023) and Ye et al. (2023) evaluate each candidate prompt using the training set and iteratively optimize the prompts based on the evaluation results. Cheng et al. (2023) train a Sequence-to-Sequence model to translate an imperfect prompt into a better one. Brooks et al. (2023) and Zhang et al. (2023c) combine reinforcement learning with prompt updating, demonstrating promising results. We extend these prompt optimization techniques to more complex interactive gaming environments, learning a robust behavioral strategy through policy-level reflection and search. Furthermore, our agent must constantly consider changes in the environment and the styles of opponents, thereby dynamically adjusting the content of the prompts.

D.5 Benchmarking LLM-based Agent

To develop an effective evaluation method for LLMs and their capabilities as agents, numerous researchers have concentrated on establishing benchmarks. SmartPlay (Wu et al., 2023c) introduces a benchmark from 6 diverse games with language descriptors for visual observation. Clembench (Chalamalasetti et al., 2023) employs Dialogue Games as testing tools, enabling rapid evaluations across a broad of models. Furthermore, Liu et al. (2023) unveil AGENTBENCH, a comprehensive benchmark that outlines eight distinct environments to assess LLMs. These benchmarks play a crucial role in evaluating both LLMs and LLM-based agents. In our future work, we aim to further evaluate Agent-Pro utilizing these benchmarks.

E Detailed Prompts

We provide detailed prompt designs for two games, including baselines and Agent-Pro in Appendix E.1, E.2, E.3 and E.4.

E.1 Baseline's Prompts For Blackjack

```
Game Rule:
### Game Rules
1. Please try to get your card total to as close to 21 as possible,
without going over, and still having a higher total than the dealer.
2. If anyone's point total exceeds 21, he or she loses the game.
3. You can only choose one of the following two actions: {"Stand", "Hit
"}. If you choose to Stand, you will stop taking cards and wait for the
dealer to finish. If you choose to Hit, you can continue to take a card,
but there is also the risk of losing the game over 21 points.
4. After all players have completed their hands, the dealer reveals their
 hidden card. Dealers must hit until their cards total 17 or higher.

Game Information:
 The dealer's face-up card is {Dealer-Card}. The dealer has another hidden
card. You don't know what it is. Your current cards are {Player-Card}.

---------- Prompt For Vanilla LLM ----------
You are a player in blackjack. Please beat the dealer and win the game.
### {Game Rules}
### {Game Information}
### Please output your action in following format: ###My action is {Your
action}, without any other text.

---------- Prompt For Radical LLM ----------
```

```
You are an aggressive player of blackjack who likes to take risks to earn
 high returns. Please beat the dealer and win the game.
### {Game Rules}
### {Game Information}
### Please output your action in following format: ###My action is {Your
action}, without any other text.

---------- Prompt For ReAct ----------
You are a player in blackjack. Please beat the dealer and win the game.
### {Game Rules}
### {Game Information}
### Please first think and reason about the current hand and then
generate your action as follows: ###My thought is {Your Thought}. My
action is {Your action}.

---------- Prompt For Reflexion ----------
You are a player in blackjack. Please beat the dealer and win the game.
### {Game Rules}
### {Game Information}
### Please first think about the current hand and then generate your
action in following format: ###My thought is {Your thought}. My action is
 {Your action}.
Assistant: {LLM Response}. My action is {LLM Response}
### Please carefully check the response you just output, and then refine
your answer. The final output is also in following format: ###My thought
is {Your thought}. My action is {Your action}.
```

E.2 Agent-Pro's Prompt For Blackjack

```
Game Rule:
### Game Rules
1. Please try to get your card total to as close to 21 as possible,
without going over, and still having a higher total than the dealer.
2. If anyone's point total exceeds 21, he or she loses the game.
3. You can only choose one of the following two actions: {"Stand", "Hit
"}. If you choose to Stand, you will stop taking cards and wait for the
dealer to finish. If you choose to Hit, you can continue to take a card,
but there is also the risk of losing the game over 21 points.
4. After all players have completed their hands, the dealer reveals their
 hidden card. Dealers must hit until their cards total 17 or higher.

Game Information:
The dealer's face-up card is {Dealer-Card}. The dealer has another hidden
card. You don't know what it is. Your current cards are {Player-Card}.

---------- Prompt For Agent-Pro ----------
You are a player in blackjack. Please beat the dealer and win the game.
### {Game Rules}
### {Game Information}
### {Behavioral Guideline: Goal, Strategy, Demonstration}
### {World Modeling: Rule Description}
### Please read the behavoiral guideline and world modeling carefully.
Then you should analyze your own cards and your strategies in Self-belief
 and then analyze the dealer cards in World-belief. Lastly, please select
 your action from {"Stand", "Hit"}.
### Output Format: Self-Belief is {Belief about youself}. World-Belief is
 {Belief about the dealer}. My action is {Your action}. Please output in
the given format.

---------- Prompt For Policy-Level Reflection ----------
### {Game Rules}
### Game Record: {Game Record, Belief Sequences, Final Result}
### You are a seasoned blackjack expert, and you need to carefully
reflect on the following record of this losing game:
```

Correctness: Whether its beliefs about yourself, the game, and the dealer align with the final results.
Consistency: Whether each belief and action is self-contradictory.
Reasons: Reflect on why you lost to your dealer, which beliefs and actions are problematic, and what the underlying reasons are.

Output Format: I analyze this game as follows: {Your analysis about the game and belief}.

---------- **Prompt For Generating Behavioral Guideline and World Modeling** ----------
Game Record: {Game Record, Belief Sequences, Final Result}
Policy-Level Reflection: {Reflection}
Following the previous rigorous analysis, you should distill and articulate a set of Behavioral Guidelines and World Modeling. The Behavioral Guideline is about what you consider to be a more reasonable and effective behavioral strategy and suggestions. World Modeling is about the description of the game and the dealer.

Here are some suggestions for you:

Behavioral Guideline
1-Goal: Please summarize the detailed goal based on your reflection ...
2-Strategy: What kind of strategy can lead you to win in similar games ...
3-Demonstration: Can this game be considered a typical example to be preserved for future reference ...
World Modeling
1-Rule-Description: Based on the recent reflection, describe any game rules or details that are easy to overlook ...

E.3 Baseline's Prompts For Limited Texas Hold'em

Game Settings:
1- You are playing the Limit Texas poker game. In this game, there are 4 players from 0 to 3, and your role is player 3.
2- The number of chips every player has is infinite.
3- You just need to win new chips in the competition as much as possible.
4- The actions you can choose are ['call', 'raise', 'fold', 'check']

Game Information:
Your current hands are {Private Cards}.
The current stage: {Stage}. Public cards are {Public Cards}.
Number of chips all players have invested are {Inveseted Chip List}.
Available actions you can choose are {Available Actions}.
Previous actions of all players are: {Preflop: Actions Sequences, Flop: Actions ...}.

---------- **Prompt For Vanilla LLM** ----------
You are a player in Limited Texas Hold'em. Beat your opponents and win the game.
{Game Rules}
{Game Information}
Output your action in following format: {"action": " "} without any other text.

---------- **Prompt For Aggressive LLM** ----------
You are an aggressive player of limited Texas Hold'em who likes to take risks to earn high returns. Please beat your opponents and win the game.
{Game Rules}
{Game Information}
Output your action in following format: {"action": " "} without any other text.

---------- **Prompt For Conservative LLM** ----------

```
You are a conservative player of limited Texas Hold'em who is risk averse
 and prefers more certainty. Please beat your opponents and win the game.
### {Game Rules}
### {Game Information}
### Output your action in following format: {"action": " "} without any
other text.

---------- Prompt For ReAct ----------
You are a player in Limited Texas Hold'em. Beat your opponents and win
the game.
### {Game Rules}
### {Game Information}
### Please first think and reason about the current state and then
generate your action as follows: ###My thought is {Your Thought}, and my
action is {"action": " "}

---------- Prompt For Reflection ----------
You are a player in Limited Texas Hold'em. Beat your opponents and win
the game.
### {Game Rules}
### {Game Information}
### Please first think and reason about the current state and then
generate your action as follows: ###My thought is {Your Thought}, and my
action is {"action": " "}.

Assistant: {LLM Response}.

### Please carefully check the thought and the action you just output,
and then refine your answer. The final output is also in the same format:
 ###My revised thought is {Your Thought}. My revised action is {"action":
 " "}.
```

E.4 AGENT-PRO'S PROMPT FOR LIMITED TEXAS HOLD'EM

```
Game Settings:
1- You are playing the Limit Texas poker game. In this game, there are 4
players from 0 to 3, and your role is player 3.
2- The number of chips every player has is infinite.
3- You just need to win new chips in the competition as much as possible.
4- The actions you can choose are ['call', 'raise', 'fold', 'check']

Game Information:
Your current hands are {Private Cards}.
The current stage: {Stage}. Public cards are {Public Cards}.
Number of chips all players have invested are {Inveseted Chip List}.
Available actions you can choose are {Available Actions}.
Previous actions of all players are: {Preflop: Actions Sequences, Flop: Actions
...}.
---------- Prompt For Agent-Pro ----------
You are a player in Limited Texas Hold'em. Beat your opponents and win
the game.
### {Game Rules}
### {Game Information}
### {Behavioral Guideline: Goal, Strategy, Demonstration}
### {World Modeling: Rule Description, Opponents Description}
### Please read the behavoiral guideline and world modeling carefully.
Then, following their instructions, you should analyze your own cards and
 your strategies in Self-belief and then analyze and reason about your
opponents in World-belief. Lastly, output your action.
### Output Format: Self-Belief is {belief about youself}. World-Belief is
 {belief about the all opponents}. My action is {"action": " "}. Please
output in the given format.
---------- Prompt For Policy-Level Reflection ----------
### {Game Rules}
```

```
### {Game Information}
### Game Record: {Game Record, Belief Sequences, Final Result}
### You are a seasoned Limited Texas Hold'em expert, and you need to
carefully reflect on the following record of this losing game:

Correctness: Whether its beliefs about yourself, the game, and the
opponents align with the final results.
Consistency: Whether each belief and action is self-contradictory.
Rationality: Whether the beliefs accurately reflect the underlying
intentions behind the opponents.
Reasons: Reflect on why you lost to your opponents, which beliefs and
actions are problematic, and what the underlying reasons are.

### Output Format: I analyze this game as follows: {Your analysis about
the game and belief}.
---------- Prompt For Generating Behavioral Guideline and World Modeling ----------
### Game Record: {Game Record, Belief Sequences, Final Result}
### Policy-Level Reflection: {Reflection}
Following the previous rigorous analysis, you should distill and
articulate a set of Behavioral Guidelines and World Modeling. The
Behavioral Guideline is about what you consider to be a more reasonable
and effective behavioral strategy and suggestions. World Modeling is
about the description of the game and all opponents.

Here are some suggestions for you:

Behavioral Guideline
1-Goal: Please summarize the detailed goal based on your reflection ...
2-Strategy: What kind of strategy can lead you to win in similar games ...
3-Demonstration: Can this game be considered a typical example to be
preserved for future reference ...
World Modeling
1-Rule-Description: Based on the recent reflection, describe any game rules
or details that are easy to overlook ...
2-Opponent-Modeling: Based on each opponent's action and the real hands
shown at the end of the game, what do you think their style and strategy
are ...
```

F CASE STUDY

As shown in Figure F3, F4, F1, F2 we provide four cases for Blackjack. We visualize the difference in their solving steps between Agent-Pro and ReAct when using Qwen-72B. Besides, we also provide four cases for Limited Texas Hold'em in Appendix F.2. These cases demonstrate that Agent-Pro, after learning, has significantly improved in understanding task rules, mastering techniques, and dealing with uncertain environments.

F.1 CASES FOR BLACKJACK

Figure F1: Some cases for Blackjack by ReAct and Agent-Pro using Qwen-72B. ReAct behavioral strategies are too aggressive. However, Agent-Pro guesses the dealer's next move and makes a reasoned decision.

State	ReAct Vs. Dealer	Agent-Pro Vs. Dealer
Initial Card	The dealer's face-up card is **Spade 4**. Your current cards are **Heart 5, Club J**.	
Thought	My thought is I currently have a total of 15 points with my two cards, which is not a very strong hand. The dealer's up-card is a 4, which is a relatively low value card. Based on basic strategy in blackjack, I should hit if my total is less than 17, so I should take a hit to improve my hand.	Self-Belief is I currently have a total of 15 points, with a Heart 5 and a Club J. I am slightly below the ideal total of 21, but I am also less likely to go bust if I draw another card. I think my best chance of winning is to stand and hope that the dealer goes bust or ends up with a lower total. World-Belief is The dealer currently has a Spade 4 showing, which means their total is likely to be between 4 and 14 points. Since the dealer must draw more cards if their total is below 17, they are at a higher risk of going bust. However, if the dealer has a strong hidden card, such as a face card or an ace, they could end up with a high total and win the game. Given this uncertainty, standing seems like a safer bet.
Action	My action is `Hit`.	My action is `Stand`.
Next Round	The final hand is Dealer: **Spade 4, Diamond 4**. Your: **Heart 5, Club J, Club 8** (23 point)	The final hand is Dealer: **Spade 4, Diamond 4, Club 8, Spade 7** (23 point). Your: **Heart 5, Club J** (15 point).
Result	Lose	Win

Figure F2: Some cases for Blackjack by ReAct and Agent-Pro using Qwen-72B. ReAct doesn't consider the risk of exceeding 21 points.

State	ReAct Vs. Dealer	Agent-Pro Vs. Dealer
Initial Card	The dealer's face-up card is **Spade 8**. Your current cards are **Heart 10, Club 8**.	
Thought	My thought is The sum of my cards is currently 18. I am in a fairly safe position, not too likely to exceed 21 by hitting. However, since the dealer's visible card is an 8, there is a good chance that the sum of the dealer's cards will also be around 18. In order to increase my chances of winning, I need to aim for a higher total.	Self-Belief is I currently have a total of 18 points, which is a relatively strong hand. I am confident that I can beat the dealer with this hand. However, I should be cautious not to exceed 21 points by hitting and risking busting. World-Belief is The dealer's visible card is an 8, which is a moderate hand. It is likely that the dealer may still need to hit to reach a total of 17 or higher. Therefore, I should be careful not to risk busting by hitting and exceeding 21 points.
Action	My action is `Hit`.	My action is `Stand`.
Next Round	The final hand is Dealer: **Spade 8, Heart 4**. (12 points) Your: **Heart 10, Club 8, Diamond J** (28 points)	The final hand is Dealer: **Spade 8, Heart 4, Diamond J** (22 points). Your: **Heart 10, Club 8** (18 points).
Result	Lose	Win

Figure F3: Some cases for Blackjack conducted by ReAct and Agent-Pro using Qwen-72B. ReAct does not accurately understand the meaning of "A" in the rules of the game.

State	ReAct Vs. Dealer	Agent-Pro Vs. Dealer
Initial Card	The dealer's face-up card is **Club 6**. Your current cards are **Club A, Diamond 4**.	
Thought	My thought is that I currently have a total of 15 points (Ace counts as 1) and the dealer has 6 points showing. To win the game, I need to get as close to 21 as possible without going over. With the dealer's current visible card being low, they may have to hit again to reach 17 or higher, increasing their chances of busting. Therefore, I should play conservatively and stand with my current total of 15 to avoid risking busting myself.	Self-Belief is I have a total of 15 points, with an Ace that can be either 1 or 11, giving me some flexibility in my strategy. I should aim to get closer to 21 without busting. World-Belief is The dealer's visible card is a 6, which means they could have a total of anywhere from 16 to 22, and may need to hit again. Given this information, I think my best strategy is to Hit and try to improve my hand.
Action	My action is `Stand`.	My action is `Hit`.
Next Round	The final hand is Dealer: **Club 6, Club 5, Heart 6**. Your: **Club A, Diamond 4**.	The dealer's face-up card is **Club 6**. Your current cards are **Club A, Diamond 4, Heart 6**
Thought	Lose Dealer: Club 6 + Club 5 + Heart 6 = 17 You: Club A + Diamond 4 = 5 or 15	Self-Belief is I currently have three cards with a total value of 21. This is a strong hand and I am confident in my chances of winning the game. World-Belief is The dealer currently has a visible card of 6, which is a relatively low value. However, I do not know the value of their hidden card, so I cannot be too complacent. Based on these factors, I believe that standing is the best course of action.
Action	Lose	My action is `Stand`.
Next Round	Lose	The final hand Dealer: **Club 6, Club 5, Diamond 9** (20). Your: **Club A, Diamond 4, Heart 6** (21).
Result	Lose	Win

Figure F4: Some cases for Blackjack by ReAct and Agent-Pro using Qwen-72B. Agent-Pro accurately understands the rules of the game and makes rational decisions.

State	ReAct Vs. Dealer	Agent-Pro Vs. Dealer
Initial Card	The dealer's face-up card is **Spade 3**. Your current cards are **Heart A, Club 9**.	
Thought	My thought is that I currently have a total of 10 points with my Heart A and Club 9. Given that the dealer's visible card is a Spade 3, it is likely that the dealer has a total of between 13 and 16 points, since they would usually stand on a hard 17 or higher. Therefore, I should aim to get as close to 21 as possible without going over, while considering the possibility that the dealer may have a strong hand.	Self-Belief is I have a total of 20 points, which is a strong hand. I am confident that I can beat the dealer with this hand. World-Belief is The dealer's visible card is a 3, which is a weak hand. However, the dealer may have a hidden card that could potentially change the outcome of the game. Taking unnecessary risks could lead to an unfavorable outcome. Therefore, I should choose to stand and wait for the dealer to reveal their hidden card.
Action	My action is `Hit`.	My action is `Stand`.
Next Round	The dealer's face-up card is **Spade 3**. Your current cards are **Heart A, Club 9, Club 5**	The final hand is Dealer: **Spade 3, Heart J, Club 5**. Your: **Heart A, Club 9**.
Thought	I have a total of 15 points from my two cards (Ace counts as 1), and the dealer shows a 3. Given this situation, there is a good chance that the dealer may bust if they draw more cards to reach at least 17. However, standing with 15 is risky as well since the dealer could still make a strong hand.	Win Heart A + Club 9 = 20 Spade 3 + Heart J + Club 5 = 18
Action	My action is `Hit`.	Win
Action	My action is `Hit`.	Win
Next Round	The dealer's face-up card is **Spade 3, Heart J, Diamond 8** (21). Your current cards are **Heart A, Club 9, Club 5, Spade K** (25 or 35)	Win
Result	Lose	Win

F.2 CASES FOR TEXAS HOLD'EM

Current game State:
Now your hand is ['HT', 'CK'], and the public cards are []. The number of chips all players have invested is [4, 0, 1, 2]. Currently, Preflop. Action Sequence: Player 0 raises. Player 1 folds. Player 2 folds.
The Response of the Agent:
Self-Belief: As Player 3, I hold the Ten of Hearts and the King of Clubs in hand, indicating a high card potential. However, no similar suits or consecutive cards suggest a weak hand in terms of potential straight or flush. Nevertheless, the hand still has some promising value given the presence of a King. Even though the community cards have not yet been dealt, it's worth to consider a strategy of maintaining some aggression to minimize the number of opponents in the later stages.
World-Belief: Player 0 seemingly adopted an aggressive strategy and raised already, potentially suggesting a strong hand, although it might also be a bluff. ··· However, whether Player 0 continues his aggressive strategy need to be closely watched. ···
Action: Given the current situation, I am not in a position to take an aggressive action, neither it is wise to fold at this point. Thus, I will choose to simply match the current bet to see the flop. Thus, my action would be **Call**.
----------**Next Round**----------
Current game State: Now your hand is ['HT', 'CK'], and the public cards are []. The number of chips all players have invested is [4, 0, 1, 4]. Currently, Preflop. Action Sequence: Player 0 raises. Player 1 folds. Player 2 folds. Player 3 calls. Player 0 checks.
The Response of the Agent:
Self-Belief: As Player 3, my hand consists of the Ten of Hearts and the King of Clubs. These are not strong cards, and they do not form any promising hands such as a straight or flush. However, the King card is high rank and may provide some advantages in the later stage of the game. A conservative strategy might be the most reasonable at this stage, but a certain level of aggression may also be necessary to exert pressure on my opponent.
World-Belief: Player 0 appears to be employing an aggressive strategy, having raised even before any community cards have been dealt. This behavior suggests that he may have a strong hand. However, it could also be possible that Player 0 is bluffing. Since players 1 and 2 have already folded, it appears that the game might turn into a head-on confrontation between Player 0 and me. ···
Action: Given this scenario, it would be wise to use a balanced strategy to counter Player 0's aggressive play without risking too many chips on my not-so-strong hand. Therefore, I will choose {'action': **'check'**}.
----------**Next Round**----------
Current game
State: Now your hand is ['HT', 'CK'], and the public cards are ['D7', 'C5', 'C4']. The number of chips all players have invested is [4, 0, 1, 4]. Currently, Preflop. Player 0 raises. Player 1 folds. Player 2 folds. Player 3 calls. Player 0 checks. Player 3 checks. Flop. Player 0 checks.
The Response of the Agent:
Self-Belief: As Player 3, the cards in my hand are the Ten of Hearts and the King of Clubs. Without a clear pattern or majority suit, my current hand doesn't offer a lot in terms of immediate strength. However, the King is a high-ranking card and could potentially provide a strategic advantage should the game progress towards a showdown. Given the moderateness of my hand and the absence of an apparent competitive advantage, it might be wise to maintain a balanced strategy, toggling between conservative and aggressive plays based on the situation.
World-Belief:Now, looking at Player 0, who remains the only player I would engage in this round, following the withdrawal of Players 1 and 2. In the series of events leading up to the Flop, Player 0 has exhibited an aggressive approach, choosing to raise at every opportunity. This could

indicate the strength of his hand, or it could be an attempt to bluff, setting up an environment where he may feed on the hesitance of the other players. Given this behavior, it seems logical to assume that he will continue this aggressive strategy going forward.···"
Action: Given the current circumstances, a cautious but unexpected move on my part could potentially throw off Player 0's supposed
strategy. Thus, I will go with {'action': 'raise'}. This could make him reconsider my playing style and the strength of my hand, potentially inducing unforeseen errors in future rounds.
----------**Next Round**----------
Player 0 fold
----------**Game End**----------
Player0 loses 2 points, Player1 loses 0, Player2 loses 0.5, Agent-Pro wins 2.5 points
----------Game End----------
Baseline Action Sequence:
Preflop: raises -> calls -> calls -> checks.
Flop: checks -> checks
Turn: calls -> raises ->raises -> raises -> checks.
River: checks -> checks.
Baseline loses by 13.0 chip(s).

Current game State:
Now your hand is ['HK', 'H2'], and the public cards are []. The number of chips all players have invested is [2, 4, 4, 2].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 calls.
The Response of the Agent:
Self-Belief: As Player 3, my card combination is currently off-suit high-low hand with King and 2 in Hearts. This is a decent beginning at the Preflop stage, especially considering the high card, King, which carries significant weight. However, it's still in the early stages of the game and no community cards are revealed, thus full information is not available to make a precise decision.
World-Belief: Player 0, Player 1, and Player 2 have all chosen to remain in the game. Player 1 has raised the pot, indicating they could potentially hold a strong hand. Player 0 and Player 2 have both elected to call, suggesting they might be playing more cautiously, possibly because they hold mediocre hands or they're waiting to see the upcoming community cards before playing aggressively. ···
Action: Given the present circumstances, I believe the smart play would be to call. While the high King gives my hand some potential, the accompanying 2 substantially weakens my initial stance. Meanwhile, an overly aggressive play could be detrimental if my opponents are holding strong hands, which is possible given Player 1's raise. Hence, my move would be to call and wait for the flop's revelation. Thus, I'd conclude with {'action': **'call'**}.
----------**Next Round**----------
Current game State:
Now your hand is ['HK', 'H2'], and the public cards are []. The number of chips all players have invested is [6, 8, 4, 4].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 calls. Player 3 calls. Player 0 raises. Player 1 raises. Player 2 folds.
The Response of the Agent:
Self-Belief: As Player 3, I currently hold a high card King and a low card 2, both in hearts. This off-suit combination is an okay starting point in the preflop round. I have to keep in mind that luck plays a role in the rounds to come and the community cards to be drawn
yet. Moreover, an aggressive approach towards the end can tilt the scales towards me, deploying my king card, if it finally pairs up.
World-Belief: ···
Action: Considering these observations and trying to keep up the misleading image, I think the best move for now would be to call, not showing too much aggression, yet staying in the game, waiting for a potentially favorable turn or river. Therefore, I will choose {'action': **'call'**}.

----------**Next Round**----------
Current game State:
Now your hand is ['HK', 'H2'], and the public cards are ['DA', 'DK', 'SK'']. The number of chips all players have invested is [8, 8, 4, 8].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 calls. Player 3 calls. Player 0 raises. Player 1 raises. Player 2 folds. Player 3 calls. Player 0 calls. Player 1 checks.
The Response of the Agent:
Self-Belief: ···
World-Belief: ···
Action: Based on this analysis, I opt for {'action': **'raise'**} demonstrating a greater level of confidence in my hand and taking advantage of my opponents' potential uncertainty.
----------**Next Round**----------
Next Action Sequence: Player 0 raises. Player 1 raises. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks.
Turn, Community Cards: ['DA', 'DK', 'SK', 'C3'], Action Sequence: Player 0 raises. Player 1 calls. Player 3 raises. Player 0 raises. Player 1 calls. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks.
River, Community Cards: ['DA', 'DK', 'SK', 'C3', 'H8'], Action Sequence: Player 0 raises. Player 1 calls. Player 3 raises. Player 0 raises. Player 1 calls. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks
----------**Game End**----------
Player0 loses 24.0 points
Player1 lost 24.0 points
Player2 lose 2.0 points
Agent-Pro wins 50.0 points
----------**Baseline**: Vanilla LLM with the Same Cards and Opponents ----------
Action Sequence:
Preflop: Calls -> Folds.
Baseline loses by 2.0 points.

Current game State:
Now your hand is ['C5', 'H3'], and the public cards are []. The number of chips all players have invested is [2, 4, 1, 2].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 folds.
The Response of the Agent:
Self-Belief: Currently, I am holding 5 of Clubs and 3 of Hearts, which is a low unsuited hand. At the preflop stage, without any public cards on the table, it's a bit risky to invest heavily with such a hand. My feasible strategies include playing it safe and calling, or folding to avoid unnecessary risk. Raising would be a more aggressive move and less rational given my current hand.
World-Belief: Looking at my competitors, Player 0 has chosen to call, indicating a possibility of moderate strength in his hand. It seems he is being cautious, potentially having a mediocre hand. Player 1 raises, suggesting a high confidence in his hand, although it could also be a bluff. As Player 2 has already folded, I only have to consider Player 0 and Player 1 in terms of active competition. ···
Action: Given the above analysis, my most reasonable action at this stage would be to call, as raising with a weak hand would be an unnecessary risk, and folding might close potential chances to improve my hand with the public cards to come. Hence, I will choose {'action': **'call'**}.
----------**Next Round**----------
Current game State:
Now your hand is ['C5', 'H3'], and the public cards are []. The number of chips all players have invested is [6, 8, 1, 4].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 folds. Player 3 calls. Player 0 raises. Player 1 raises.
The Response of the Agent:
Self-Belief: Currently, I am holding 5 of Clubs and 3 of Hearts, which is a weak hand, especially in the absence of any public cards. With this hand,

my feasible strategies include calling, to maintain the game pace while minimizing potential loss, or folding to completely avoid any risk. Raising, in this scenario, seems irrational given the weak hand.
World-Belief: ···
Action: Based on these analyses, the most reasonable action would be to play it safe. Therefore, I will choose {'action': **'call'**}.
----------**Next Round**----------
Current game State:
Now your hand is ['C5', 'H3'], and the public cards are []. The number of chips all players have invested is [10, 10, 1, 8].
In Preflop, Action Sequence: Player 0 calls. Player 1 raises. Player 2 folds. Player 3 calls. Player 0 raises. Player 1 raises. Player 3 calls. Player 0 raises. Player 1 calls.
The Response of the
Agent: Currently, I am holding 5 of Clubs and 3 of Hearts, which is a weak hand, especially in the absence of any public cards. With this hand, my feasible strategies include calling, to maintain the game pace while minimizing potential loss, or folding to completely avoid any risk. Raising, in this scenario, seems irrational given the weak hand.
Self-Belief: ···
World-Belief: ···
Action: Given all the above factors, I think the most reasonable action would be to {'action': **'call'**}. Considering the infinite chips, my main goal is to stay in the game and watch how it unfolds. This strategy is also in line with my conservatism and allows me to stay in the round with minimal risk.
----------**Next Round**----------
Next Action Sequence: Player 0 raises. Player 1 calls. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks.
Turn, Community cards: ['D3', 'C3', 'HQ', 'S3'], Action Sequence: Player 0 raises. Player 1 calls. Player 3 raises. Player 0 raises. Player 1 calls. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks.
River, Community cards: ['D3', 'C3', 'HQ', 'S3', 'DA'], Action Sequence: Player 0 raises. Player 1 calls. Player 3 raises. Player 0 raises. Player 1 calls. Player 3 raises. Player 0 calls. Player 1 calls. Player 3 checks.
----------**Game End**----------
Player0 loses 25.0 points
Player1 loses 25.0 points
Player2 loses 0.5 points
Agent-Pro wins 50.5 points
----------**Baseline**: Vanilla LLM with the Same Cards and Opponents ----------
Action Sequence:
Preflop: Folds.
Baseline loses by 1.0 points..

Current game State:
Now your hand is ['HT', 'CK'], and the public cards are []. The number of chips all players have invested is [4, 0, 6, 2].
In Preflop, Action Sequence: Player 0 raises. Player 1 folds. Player 2 raises.
The Response of the Agent:
Self-Belief: My hand cards are heart-ten and club-king which is considered as high cards. ··· My current strategy can be to try and match my King or Ten with the upcoming community cards or use my King card as the high card to potentially outrank my opponents.
World-Belief: ··· Their betting strategy could be quite aggressive which can be impactful in trying to potential opponents to fold, allowing them to scoop the pot or they have strong confidence in their hands. ···
Action: ··· I will choose {'action': **'call'**}.
----------**Next Round**----------
Next Action Sequence: Player 0 raises. Player 2 calls. Player 3 calls. Player 0 checks.
Flop, community cards is ['D7', 'C5', 'C4'], Action Sequence: Player 2 checks.

----------**Next Round**----------
Current game State:
Now your hand is ['HT', 'CK'], and the public cards are ['D7', 'C5', 'C4 ']. The number of chips all players have invested is [8, 0, 8, 8].
In Preflop, Action Sequence: Player 0 raises. Player 1 folds. Player 2 raises. Player 3 calls. Player 0 raises. Player 2 calls. Player 3 calls. Player 0 checks.
In Flop, Action Sequence: Player 2 checks.
The Response of the Agent:
Self-Belief: ··· Based on the Texas Hold'em hand ranking, I only have a high card at this moment. Nonetheless, if the Turn and the River bring clubs, I could potentially form a Flush with my King as the highest card. But, the probability of this happening is relatively low, and relying solely on this possibility would be too risky and against the prudent poker strategy.
World-Belief: ···
Action: ··· So, {'action': **'check'**} is my choice.
----------**Next Round**----------
Next Action Sequence: Player 0 checks. Player 2 raises. Player 3 calls. Player 0 folds. Player 2 raises. Player 3 calls. Player 2 checks. Player 3 checks.
Turn, community cards: ['D7', 'C5', 'C4', 'C3'], Action Sequence: Player 2 checks. Player 3 checks. Player 2 checks. Player 3 checks.
----------**Next Round**----------
Current game State:
Now your hand is ['HT', 'CK'], and the public cards are ['D7', 'C5', 'C4 ', 'C3', 'D2']. The number of chips all players have invested is [8, 0, 16, 12].
In Preflop, Action Sequence: Player 0 raises. Player 1 folds. Player 2 raises. Player 3 calls. Player 0 raises. Player 2 calls. Player 3 calls. Player 0 checks.
In Flop, Action Sequence: Player 2 checks. Player 3 checks. Player 0 checks. Player 2 raises. Player 3 calls. Player 0 folds. Player 2 raises. Player 3 calls. Player 2 checks. Player 3 checks.
In Turn, Action Sequence: Player 2 checks. Player 3 checks. Player 2 checks. Player 3 checks.
In River, Action Sequence: Player 2 raises.
The Response of the Agent:
Self-Belief: As Player 3, my hand consists of the ten of hearts and king of clubs. The community cards, unfortunately, don't create any viable combinations for me except for a potential straight if an Ace or 6 of any suit appears, but that's pushing my luck. My current strategy is to proceed with caution, bluff if necessary, and only call when my hand has a good potential.
World-Belief: Looking at my opponents, Player 0 and Player 2, they started strong by raising the stakes but have seemed to cool off a bit. This might signify that the community cards didn't play into their hand as they would've hoped. ···
Action: Considering all the above, I think it's vital to maintain my poker face and keep them guessing. I will {'action': **'call'**}. I will show confidence while remaining thoughtful of the game situation.
----------**Next Round**----------
Next Action Sequence: Player 2 checks. Player 3 checks.
----------**Game End**----------
Player0 loses 4.0 points
Player1 loses 0.0 points
Player2 wins 12.0 points
Agent-Pro loses 8.0 points
----------**Baseline**: Vanilla LLM with the Same Cards and Opponents ----------
Action Sequence:
1-Preflop: raises -> calls. 2-Flop: checks. 3-Turn: checks -> calls.
4-River: checks -> calls -> calls -> raises -> raises -> checks.
Player 3 loses by 15.0 points.