PLAYER*: Enhancing LLM-based Multi-Agent Communication and Interaction in Murder Mystery Games

Qinglin Zhu[1*], **Runcong Zhao**[1*], **Jinhua Du**[2], **Lin Gui**[1], **Yulan He**[1,3]
[1]King's College London, [2]Huawei London Research Centre, [3]The Alan Turing Institute
{qinglin.1.zhu,runcong.zhao}@kcl.ac.uk
jinhua.d@huawei.com,{lin.1.gui, yulan.he}@kcl.ac.uk

Abstract

Recent advancements in Large Language Models (LLMs) have enhanced the efficacy of agent communication and social interactions. Despite these advancements, building LLM-based agents for reasoning in dynamic environments involving competition and collaboration remains challenging due to the limitations of informed graph-based search methods. We propose PLAYER*, a novel framework based on an anytime sampling-based planner, which utilises sensors and pruners to enable a purely question-driven searching framework for complex reasoning tasks. We also introduce a quantifiable evaluation method using multiple-choice questions and construct the WellPlay dataset with 1,482 QA pairs. Experiments demonstrate PLAYER*'s efficiency and performance enhancements compared to existing methods in complex, dynamic environments with quantifiable results.

1 Introduction

Recent advancements in LLMs capable of generating human-like responses have boosted the development of LLM-as-Agent. Building upon this progress, a series of studies focusing on multi-agent communications have showcased the emergence of social interactions, including cooperation (Li et al., 2024a; FAIR et al., 2022), trust (Xu et al., 2023a), deception (Wang et al., 2023), and the spread of information (Park et al., 2023). Despite notable progress in enabling LLM-based agents to mimic human language, building agents for reasoning in dynamic environments, especially in scenarios considering competition and collaboration with other agents, remains a challenge, for example, the Murder Mystery Games (MMGs), a strategic game requires both cooperation and competition among 4-12 players through negotiation and tactical coordination (Figure 1). We argue that the primary reason is that most existing LLM-based agents involve the informed graph-based search method with a heuristics term to guide the reasoning progress. While this approach might be efficient in well-defined problems, where constructing a relation graph and producing a clear reasoning path as output is feasible, in complex problems requiring both cooperation and competition through natural language negotiation and tactical coordination, defining such a clear graph of potential actions becomes challenging. The main difficulty of applying traditional Multi-Agent Reinforcement Learning (MARL) methods in such scenarios arises from the challenges in defining state spaces, action spaces, and rewards, where the reward associated with the final decision often represents the only clear certainty (FAIR et al., 2022; Shi et al., 2024).

Therefore, we propose a new framework based on an anytime sampling-based planner. Unlike informed graph-based search methods, such as A* (Dechter & Pearl, 1985), the anytime sampling-based planner, such as RRT* (Karaman & Frazzoli, 2011) or BIT* (Gammell et al., 2020a), tackles optimal motion planning problems in a more complex scenario. That is, by sampling possible states with detective sensors in real-time, the tree-style search path is constructed and pruned dynamically until the target is reached. Similarly, in MMGs, by proposing a set of sensors considering relations and emotions between players during interaction, along with a pruner that targets extracting highly suspicious murders, we

[*]Equal contribution.

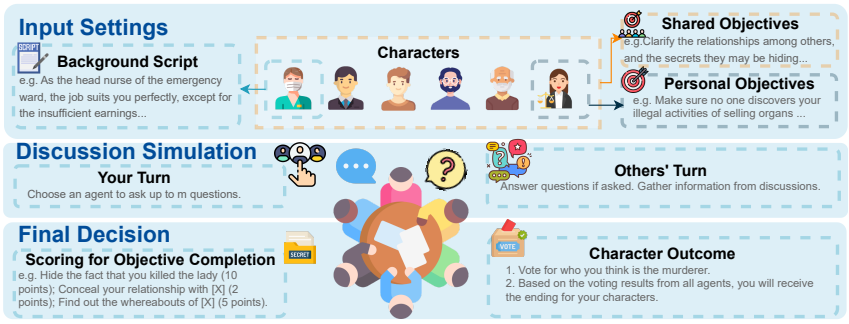

Figure 1: The gameplay comprises three stages: introduction, discussion, and decision. During the introduction stage, players familiarise themselves with their characters' background information and objectives, then briefly introduce themselves in character. Subsequently, they engage in interactive discussions to deduce the identity of the murderer.

construct a purely questioning-driven search framework. Our main contribution is that we proposed an approach which constructs a reasoning process without the need for predefined questions or prompt templates. Instead, it relies on a generator regulated by sensors and a pruner to efficiently generate high-quality questions for the agent's reasoning task.

Moreover, the current evaluation is difficult to quantify. These methods rely heavily on manual evaluation for in-depth understanding but are considerably influenced by subjective interpretation (Xu et al., 2023a; Wang et al., 2023).To address this, we propose a quantifiable and reproducible evaluation criterion by devising a series of multiple-choice questions. This approach encompasses three types of questions: fact-based questions that are automatically generated from an existing dataset focusing on character relationships (Zhao et al., 2024c), questions about the shared and individual objectives of each character set within the game, and questions probing the rationale behind the given responses.

In summary, we made the following contributions: (1) We propose PLAYER*, a framework aimed at efficiently optimising path planning in MMGs. (2) We conduct experiments to assess the efficiency and performance enhancements brought by PLAYER*, demonstrating its superiority over existing multi-agent methods. (3) We propose a quantifiable evaluation method using multiple-choice questions focusing on facts, character objectives, and reasoning, mitigating the subjectivity in current evaluation practices with the construction and annotation of the corresponding dataset, WellPlay, which includes 1,482 QA pairs. [1].

2 PLAYER*

2.1 Problem Setting & Preliminary: Search by LLMs

In response to the complexities of social interactions such as MMGs, we have developed an innovative interactive framework tailored for such scenarios. As illustrated in Figure 1, this framework entails the creation of a set of agents $\mathcal{A} = \{a_i\}_{i=1}^{N_a}$ for a game featuring N_a playable characters. $\mathcal{V} = \{v_i\}_{i=1}^{N_v}$ represents the set of victims in the game. Each agent a_i is assigned to a corresponding character and initialised with its role background script C_i, suspicious state $s_i = [s_{ik}]_{k=1}^{N_v}$, and objectives o_i. Here, C_i is designed from the unique viewpoint of that character, framing all relationships and events within the story from a_i's perspective. The suspicious state s_{ik} is a vector representing the suspicions of killing victim v_k from the perspective of agent a_i, which is an $(N_a - 1)$-dim vector, where the entries can be discrete or continuous based on suspicion strength. The complete game rules and examples can be found in the appendix A.

[1]Our code and dataset are available at `https://github.com/alickzhu/PLAYER`

Unlike transitional planning task in which the state space can be accessed directly, s_i in our task is generated from a language model. Therefore, searching in state space is done by adjusting the prompting. We define the possible prompts as an action space, allowing the states to be obtained through an LLM-based search: $s_i = P(C_i', T)$, where P is the LLM, $C_i' \subseteq C_i$ represents the related scripts, and T is the prompt instruction. The preliminary planner contains two main components:

- **Explorer**: By choosing a different prompt template T, which is usually from a pre-defined prompt template set, explore the corresponding s_i.
- **Predictor**: a self-reflection estimator to evaluate s_i. collaborate with other heuristic strategies (like A* or MC tree search) to determine if it is necessary to explore another s_i by switching T, or using this s_i and a clue to continuously explore in the next iteration.

In general, the preliminary method only construct a prompting based planner for a give script. However, in the complex scenario like MMGs, the objective can be multiple, noted as $\mathbf{o}_i = \{o_{ij}\}_{j=1}^{N_i}$, including both personal and shared objectives. Subsequently, the agents requires to engage in a simulated discussion by asking questions and expressing opinions. To ensure fairness, the game imposes a limit of m questions per agent per round, spanning a total of n rounds. Following these rounds, agents are required to make decisions and cast their votes to identify the suspected murderer. Therefore, we need to construct a more complicated strategies considering the simulation among agents.

2.2 Construction of Agents

The strength of the preliminary method is that the tree structure and the potential reasoning path is clearly defined by the T. However, the lack of flexibility and the over simplified explorer may lead to a over consistency result and limit the reasoning capability of LLM. But the entirely free generation, especially allowing the communication between agents may lead to an uncontrollable results. Therefore, in our framework, we propose another two components to regularise the generation of T before and after prompting respectively:

- **Sensors**: inspired by sociology, which claims that the interpersonal relationships are often conceptualised within a multidimensional space encompassing factors(Latané, 1981; Zhao et al., 2021; Trope & Liberman, 2010), we propose employing a set of sensors associated with each task. For example, the emotion, motivation, and suspicion, are considered as the hints to generate prompt for state searching.
- **Pruner**: After prompting, we design a pruner to reduce the state space to only few high suspicions in the next round of reasoning.

Besides, To address the constraints of LLMs' finite context window and the performance degradation associated with an increasing number of input tokens (Liu et al., 2023a), we implement a **Memory Retrieval** module to store and retrieve narrative scripts and dialog logs generated during gameplay. Specifically, we store the embeddings of all generated dialog logs and each agent's script in a vector database dedicated to each agent. When an agent encounters a new event requiring action, we use the Faiss library (Douze et al., 2024) to retrieve relevant memories. This process enables the construction of prompts tailored to the predetermined maximum script length and dialog history length. It ensures consistent and coherent interactions among agents while assisting in their strategic planning.

2.3 PLAYER* Planning Strategy

To navigate an unknown continuous space defined by natural language, which encompasses the dynamics of relationships, perceptions towards the agent, and hidden secrets, PLAYER* approximate the search domain through sampling, and planning the shortest path to the agent's objective by prioritising searches based on the quality of potential solutions. As illustrated in Figure 2, this framework is fundamentally composed of two key components:

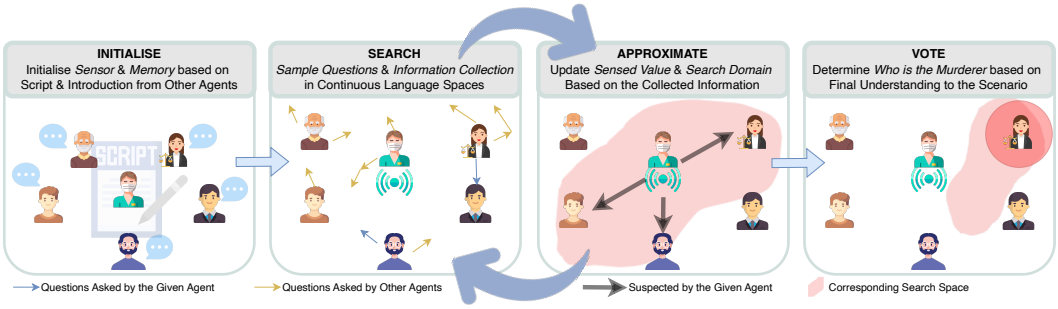

Figure 2: PLAYER*'s planning strategy begins with an introduction round, during which each agent provides a brief self-introduction. Agents then initialise scenario understanding and begin questioning other agents. Subsequently, they start searching for further information by questioning other agents. Agents then adapt their level of suspicion and intended search area based on their received response, mirroring human gameplay strategies. This iterative process of inquiry and refinement continues until the final turn, when agents make decisions aligned with their objectives.

Questioning Strategy in Searching based on Sensors Given the intricate social complexities inherent to MMGs, our proposed framework provides a robust solution through action generation with sensor components. Unlike prior approaches that required predefined prompts and a multitude of pre-set inquiries (Xu et al., 2023b), we generate actions directed towards the targets based on the values detected by those sensors. The task-specific sensors can vary across different tasks, and the only requisite input is a list of sensors. For MMGs, we have designated the input sensors S to be $[S_{Emotion}, S_{Motivation}, S_{Suspicion}]$: **Emotion**, which assesses one's willingness to assist the given individual or their intent to uncover as many of their issues as possible; **Motivation**, evaluating from one's perspective whether the individual possesses a motive to be the perpetrator; and **Suspicion**, objectively determining whether they had the opportunity to commit the crime.

Since the number of inquiries an agent can pose is limited, each question incurs a cost, representing a missed opportunity for exploration, while the reward for such a cost is the information that can be obtained from the response. Under the assumption that increased inquiries to an agent corresponds to the heightened exploration within the world space related to that agent, the expected reward associated with questioning the same agent decreases in proportion to the number of inquiries posed. Therefore, it is necessary to evaluate the probability of obtaining valuable information by persisting in questioning a character who has already been asked many questions, even if this person might be highly suspicious. We estimate this probability $P_{informative}(D_i, |Q_i|)$ using the previous dialog D_i about the agent to be questioned, as well as the number of inquiries that have already posed $|Q_i|$. Specifically, for agent a_i to be questioned, PLAYER* would generate the question q using the values of sensors S_i in conjunction with the expected reward $P_{informative}$ and other related information (C_i', D_i), that is $f_Q : (C_i', D_i, S_i) \rightarrow q$.

Action Space Refinement with a Pruner Following the search step, agents received the response from the questioned agent, as well as conversational exchanges among other agents. Each agent first updates its comprehension of the interpersonal relationships, thereby adjusting sensor values: $f_{sensor} : (C_i', D_i) \rightarrow S$. Subsequently, it also updated $P_{informative}$ based on the updated information (S_i, D_i) and the changes in sensor values ΔS_i.

In human gameplay, after an initial round of discussion, players often formulate judgments regarding the scenario and direct their focus towards a subset of highly suspicious individuals. This strategic refinement process is critical for conserving cognitive resources and enhancing the efficacy of inferential reasoning. Emulating the cognitive pattern of human beings, we introduce an Action Space Refinement strategy to refine the search space $f_{approximate} : (\mathcal{A}, S_i, P_{informative}) \rightarrow s_i$: guided by sensors, the agent selects the most

Algorithm 1: Anytime Sampling-based Planning in Multi-agent Interaction

Input: Agents $\mathcal{A} = \{a_i\}_{i=1}^{N_a}$, Victims $\mathcal{V} = \{v_i\}_{i=1}^{N_v}$, Suspicious States s, Max Round n
Output: Evaluation of Results

current_round = 0
while *current_round* < n **do**
    // Action Generation
    for i = *1 to* N_a **do**
        for k = *1 to* N_v **do**
            $suspect\_list_{ik}$ = Suspect_Generation(s_{ik})
            for a_j *in* $suspect\_list_{ik}$ **do**
                question = Action_Generation(a_i, a_j, v_i)
                answer = Reply(a_i)
    // Constrain action space
    for i = *1 to* N_a **do**
        for k = *1 to* N_v **do**
            s_{ik} = Refinement_Action_Space(a_i, a_j, v_i)
    current_round += 1
// Evaluation
for i = *1 to* N_a **do**
    Result = Evaluate(a_i)

suspicious targets. We did not impose a hard restriction on the refinement process and allowed for flexible adjustment of the action space. For instance, the number of suspects could increase, or an agent might shift suspicion to someone previously overlooked. This is because new information might entirely overturn conclusions drawn in prior rounds, and we did not want to limit that possibility. This approach prevents premature convergence on a suspect and ensures a comprehensive evaluation of all potential leads. By adopting this protocol, agents systematically converge on the most plausible suspects, thereby expediting the resolution of the mystery and heightening the likelihood of identifying the perpetrator within the designated number of discussion rounds.

Algorithm 1 outlines a detailed procedure of the game process. While initially designed for MMGs, the framework's adaptable nature allows it to be readily applied to various games by adjusting the game rules and task-specific sensors, thereby enhancing the gaming experience and offering a structured way for agents to handle complex social dynamics efficiently.

3 Experiments

3.1 Experimental Setup

Base Models We conducted experiments with GPT-3.5 for conversation and the GPT Embedding Model for memory retrieval via the Azure API in Jan-Mar 2024, using the default model versions gpt-35-turbo-16k 0613 and text-embedding-ada-002. To minimise randomness, we conducted the evaluation experiments 3 times and report the average and variance. The detailed introduction to the experiment settings is in Appendix C.1.

Baselines For baselines, we compare our approach with other multi-agent algorithms designed for multiplayer deduction games, such as Werewolf (Xu et al., 2023a), Objective-Guided Chain of Thought (O-CoT) (Park et al., 2023; Zhao et al., 2024b), and ThinkThrice (Wu et al., 2024). Additionally, we assess the performance of agents that do not actively participate in the game but have direct access to either their own script (Single Script Access) or the scripts of all agents (Full Script Access). In werewolf, questions are chosen from a role-specific predefined list to facilitate game progression, alongside questions generated based on the current scenario. In O-CoT, agent interactions are driven by their set objectives. ThinkThrice has its agents crafting questions from retrieved memory and the current scenario. However, the efficiency of exploration with the questioning strategies is low, as evidenced by their performance, which falls significantly short of full script access.

In contrast, our method leverages sensor data to identify the optimal domain to question and guide the generation of questions. The detailed comparison is in Appendix C.2.

WellPlay Dataset We create an evaluation dataset, called the WellPlay Dataset, based on the existing dataset Conan (Zhao et al., 2024c), originated from background narratives created for MMGs, which provides annotated relationships between characters. We aim to concentrate on the most challenging and significant relationships that assess the agents' comprehension of the scenario. Our approach is grounded in the original game's objectives and informed by a trial round involving human players. This method enabled us to formulate the following types of evaluation questions in line with the human players:

1. Objective. Including shared objectives, such as identifying the perpetrator(s), and individual objectives, such as determining who stoles wallet, for each character in the game.

2. Reasoning. This entails questions that delve into the reasoning behind provided answers, relating to agents' objectives, including: Who; What (the nature of the incident, such as murder, theft, disappearance, explosion); When (the time of the incident); Where(the location of the incident); Cause (e.g., shooting, poisoning, stabbing); Motive (e.g., crime of passion, vendetta, financial conflicts, manslaughter).

3. Relations. This includes interpersonal relationships between victims and other characters, as well as relationships among suspects.

To establish a quantifiable evaluation method, we employ *multiple-choice questions* focusing on *factual information*. This dataset encompasses 12 MMGs, comprising a total of 1,482 evaluation questions. On average, each game features 5.67 agents and 1.75 victims. More detailed statistics are available in the appendix B.1.

Evaluation Metrics We adopted the scoring system used in the game to evaluate agents' performance: Awarding 10 points for achieving an objective, 5 points for correct reasoning, and 2 points for providing additional information based on the players' understanding of the current situation. However, instead of using open-ended questions with human players, we assess each agent's performance using our constructed dataset, WellPlay, to facilitate replication and comparison for future research. WellPlay comprises three types of questions: objective, reasoning, and relations, which are assigned 10, 5, and 2 points, respectively. We calculate the agent's final score as $\frac{\text{Awarded Points}}{\text{Total Points}}$ to determine its overall performance. Additionally, we assess the accuracy of each question type. For the computation of the overall score for performance assessment, we have utilized weighted mean and weighted standard deviation methodologies. The detailed algorithm for these calculations is in Appendix B.2.

3.2 Results

Performance Evaluation Results Table 1 presents the evaluation of agents' performance across 12 unique games of varying complexity and settings. The number of evaluation questions ("#QA") varies based on script complexity, with more complex scripts generating a larger volume of questions. The SSA (Single Script Access) measures agents' performance with access to only their own script. This setup is designed to represent the starting point for searching. The FSA (Full Script Access) measures performance when agents have access to all agents' scripts, representing the ideal search endpoint. Ideally, FSA should aim to achieve an accuracy of 1. However, in practice, its effectiveness is often limited by the deductive capabilities of the underlying base model. Despite this, FSA still scores significantly higher than SSA. In addition, FSA also serves as an indicator of the complexity of the script.

We can observe that PLAYER* exhibits superior performance other baselinse across all evaluation questions, demonstrating its enhanced understanding of the search space through interactions with other agents. Additionally, PLAYER* significantly outperforms others in objective questions, even surpassing FSA, which has access to all information. This showcases the effectiveness of PLAYER* in refining information and dynamically narrowing down the search domain to achieve the target objective. Despite some baselines exhibiting

competitive performance in reasoning or relation questions, their significant drop in performance concerning objectives indicates a critical limitation: an inability to effectively utilise the collected information to reach correct conclusions or achieve game-specific goals.

We also evaluate the performance of our framework on English scripts in Appendix C.3 and open-source LLMs in Appendix C.4. The detailed dialogue history and evaluation records are available in the GitHub link provided previously.

Script	Evaluation	#QA	SSA	FSA	Agent's Response After Playing the Game			
					Werewolf	O-CoT	ThinkThrice	PLAYER*
Death Wears White *(9 players, 1 victim)*	Objective	10	$.233_{\pm.058}$	$.467_{\pm.116}$	$.333_{\pm.058}$	$.267_{\pm.058}$	$.267_{\pm.058}$	$.267_{\pm.116}$
	Reasoning	102	$.454_{\pm.012}$	$.503_{\pm.015}$	$.350_{\pm.021}$	$.395_{\pm.031}$	$.392_{\pm.010}$	$.408_{\pm.030}$
	Relations	72	$.431_{\pm.036}$	$.425_{\pm.031}$	$.454_{\pm.035}$	$.472_{\pm.037}$	$.398_{\pm.032}$	$.500_{\pm.048}$
	Overall	184	$.420_{\pm.007}$	$.480_{\pm.011}$	$.367_{\pm.015}$	$.393_{\pm.009}$	$.377_{\pm.013}$	$.407_{\pm.007}$
Ghost Revenge *(7 players, 3 victims)*	Objective	19	$.333_{\pm.030}$	$.334_{\pm.132}$	$.368_{\pm.052}$	$.316_{\pm.190}$	$.404_{\pm.080}$	$.509_{\pm.133}$
	Reasoning	152	$.390_{\pm.020}$	$.533_{\pm.030}$	$.417_{\pm.023}$	$.410_{\pm.019}$	$.450_{\pm.056}$	$.399_{\pm.040}$
	Relations	69	$.246_{\pm.014}$	$.411_{\pm.103}$	$.348_{\pm.014}$	$.309_{\pm.047}$	$.420_{\pm.038}$	$.386_{\pm.030}$
	Overall	240	$.362_{\pm.016}$	$.483_{\pm.020}$	$.399_{\pm.009}$	$.381_{\pm.049}$	$.438_{\pm.030}$	$.417_{\pm.032}$
Danshui Villa *(7 players, 2 victims)*	Objective	12	$.389_{\pm.048}$	$.222_{\pm.127}$	$.167_{\pm.000}$	$.305_{\pm.048}$	$.306_{\pm.096}$	$.361_{\pm.127}$
	Reasoning	128	$.315_{\pm.020}$	$.422_{\pm.016}$	$.307_{\pm.044}$	$.352_{\pm.008}$	$.347_{\pm.005}$	$.367_{\pm.008}$
	Relations	63	$.323_{\pm.040}$	$.476_{\pm.028}$	$.344_{\pm.051}$	$.381_{\pm.069}$	$.333_{\pm.016}$	$.370_{\pm.056}$
	Overall	203	$.326_{\pm.014}$	$.403_{\pm.028}$	$.293_{\pm.040}$	$.349_{\pm.012}$	$.339_{\pm.013}$	$.367_{\pm.011}$
Unfinished Love *(7 players, 2 victims)*	Objective	12	$.167_{\pm.000}$	$.167_{\pm.083}$	$.167_{\pm.144}$	$.278_{\pm.048}$	$.111_{\pm.127}$	$.222_{\pm.127}$
	Reasoning	61	$.552_{\pm.025}$	$.563_{\pm.025}$	$.563_{\pm.010}$	$.486_{\pm.025}$	$.552_{\pm.019}$	$.601_{\pm.019}$
	Relations	72	$.500_{\pm.028}$	$.546_{\pm.021}$	$.523_{\pm.071}$	$.583_{\pm.028}$	$.556_{\pm.042}$	$.616_{\pm.016}$
	Overall	145	$.457_{\pm.017}$	$.475_{\pm.033}$	$.469_{\pm.018}$	$.467_{\pm.004}$	$.460_{\pm.030}$	$.525_{\pm.029}$
Cruise Incident *(5 players, 1 victim)*	Objective	4	$.167_{\pm.289}$	$.250_{\pm.250}$	$.500_{\pm.250}$	$.500_{\pm.000}$	$.583_{\pm.382}$	$.667_{\pm.144}$
	Reasoning	24	$.653_{\pm.064}$	$.472_{\pm.064}$	$.542_{\pm.042}$	$.528_{\pm.024}$	$.528_{\pm.105}$	$.514_{\pm.024}$
	Relations	30	$.267_{\pm.058}$	$.411_{\pm.102}$	$.456_{\pm.020}$	$.456_{\pm.051}$	$.422_{\pm.039}$	$.434_{\pm.058}$
	Overall	58	$.459_{\pm.012}$	$.415_{\pm.079}$	$.510_{\pm.037}$	$.503_{\pm.017}$	$.509_{\pm.107}$	$.520_{\pm.025}$
Sin *(4 players, 1 victim)*	Objective	3	$.000_{\pm.000}$	$.333_{\pm.334}$	$.111_{\pm.192}$	$.111_{\pm.192}$	$.111_{\pm.192}$	$.333_{\pm.334}$
	Reasoning	20	$.483_{\pm.058}$	$.567_{\pm.029}$	$.767_{\pm.058}$	$.600_{\pm.100}$	$.717_{\pm.058}$	$.600_{\pm.000}$
	Relations	21	$.476_{\pm.082}$	$.587_{\pm.099}$	$.429_{\pm.000}$	$.699_{\pm.055}$	$.524_{\pm.048}$	$.698_{\pm.027}$
	Overall	44	$.397_{\pm.029}$	$.531_{\pm.053}$	$.570_{\pm.000}$	$.539_{\pm.029}$	$.564_{\pm.012}$	$.577_{\pm.053}$
Deadly Fountain *(4 players, 1 victim)*	Objective	3	$.000_{\pm.000}$	$.556_{\pm.193}$	$.000_{\pm.000}$	$.111_{\pm.192}$	$.000_{\pm.000}$	$.111_{\pm.192}$
	Reasoning	21	$.413_{\pm.153}$	$.667_{\pm.126}$	$.460_{\pm.027}$	$.476_{\pm.082}$	$.539_{\pm.055}$	$.587_{\pm.028}$
	Relations	12	$.305_{\pm.048}$	$.389_{\pm.048}$	$.333_{\pm.084}$	$.222_{\pm.096}$	$.222_{\pm.048}$	$.361_{\pm.127}$
	Overall	36	$.319_{\pm.097}$	$.604_{\pm.104}$	$.354_{\pm.022}$	$.369_{\pm.053}$	$.390_{\pm.043}$	$.463_{\pm.041}$
Unbelievable Incident *(5 players, 1 victim)*	Objective	4	$.083_{\pm.144}$	$.083_{\pm.144}$	$.333_{\pm.144}$	$.167_{\pm.144}$	$.083_{\pm.144}$	$.250_{\pm.000}$
	Reasoning	24	$.389_{\pm.105}$	$.528_{\pm.064}$	$.431_{\pm.064}$	$.361_{\pm.064}$	$.264_{\pm.127}$	$.583_{\pm.042}$
	Relations	15	$.422_{\pm.102}$	$.822_{\pm.102}$	$.644_{\pm.077}$	$.756_{\pm.102}$	$.289_{\pm.154}$	$.556_{\pm.077}$
	Overall	43	$.330_{\pm.081}$	$.481_{\pm.020}$	$.444_{\pm.037}$	$.382_{\pm.035}$	$.230_{\pm.034}$	$.509_{\pm.038}$
Desperate Sunshine *(4 players, 1 victim)*	Objective	3	$.333_{\pm.000}$	$.556_{\pm.193}$	$.556_{\pm.193}$	$.333_{\pm.000}$	$.667_{\pm.334}$	$.778_{\pm.192}$
	Reasoning	18	$.556_{\pm.056}$	$.778_{\pm.096}$	$.648_{\pm.032}$	$.704_{\pm.032}$	$.759_{\pm.064}$	$.741_{\pm.085}$
	Relations	36	$.537_{\pm.064}$	$.611_{\pm.048}$	$.556_{\pm.028}$	$.630_{\pm.043}$	$.500_{\pm.056}$	$.565_{\pm.070}$
	Overall	57	$.514_{\pm.013}$	$.680_{\pm.047}$	$.599_{\pm.023}$	$.618_{\pm.020}$	$.647_{\pm.075}$	$.680_{\pm.047}$
Riverside Inn *(4 players, 1 victim)*	Objective	3	$.444_{\pm.193}$	$.889_{\pm.192}$	$.444_{\pm.193}$	$.000_{\pm.000}$	$.667_{\pm.000}$	$.556_{\pm.193}$
	Reasoning	18	$.519_{\pm.032}$	$.667_{\pm.056}$	$.463_{\pm.064}$	$.500_{\pm.056}$	$.685_{\pm.032}$	$.667_{\pm.056}$
	Relations	18	$.407_{\pm.085}$	$.500_{\pm.096}$	$.463_{\pm.085}$	$.426_{\pm.064}$	$.389_{\pm.000}$	$.426_{\pm.032}$
	Overall	39	$.479_{\pm.032}$	$.671_{\pm.036}$	$.459_{\pm.073}$	$.387_{\pm.020}$	$.614_{\pm.018}$	$.590_{\pm.023}$
Solitary Boat Firefly *(6 players, 4 victims)*	Objective	20	$.267_{\pm.104}$	$.500_{\pm.050}$	$.183_{\pm.058}$	$.350_{\pm.132}$	$.217_{\pm.076}$	$.517_{\pm.104}$
	Reasoning	109	$.318_{\pm.014}$	$.569_{\pm.018}$	$.370_{\pm.014}$	$.425_{\pm.046}$	$.401_{\pm.014}$	$.379_{\pm.005}$
	Relations	69	$.483_{\pm.017}$	$.507_{\pm.014}$	$.536_{\pm.014}$	$.565_{\pm.038}$	$.503_{\pm.044}$	$.594_{\pm.087}$
	Overall	198	$.332_{\pm.015}$	$.544_{\pm.024}$	$.354_{\pm.023}$	$.430_{\pm.016}$	$.375_{\pm.015}$	$.444_{\pm.030}$
Manna *(6 players, 3 victims)*	Objective	24	$.250_{\pm.042}$	$.389_{\pm.086}$	$.403_{\pm.064}$	$.444_{\pm.064}$	$.430_{\pm.087}$	$.569_{\pm.105}$
	Reasoning	123	$.458_{\pm.005}$	$.393_{\pm.040}$	$.569_{\pm.028}$	$.499_{\pm.033}$	$.526_{\pm.017}$	$.539_{\pm.021}$
	Relations	88	$.447_{\pm.058}$	$.640_{\pm.036}$	$.538_{\pm.043}$	$.614_{\pm.030}$	$.511_{\pm.034}$	$.580_{\pm.046}$
	Overall	235	$.408_{\pm.017}$	$.434_{\pm.002}$	$.525_{\pm.017}$	$.505_{\pm.004}$	$.501_{\pm.025}$	$.553_{\pm.045}$
Overall	Objective	117	$.256_{\pm.121}$	$.370_{\pm.184}$	$.293_{\pm.157}$	$.333_{\pm.140}$	$.328_{\pm.187}$	$\mathbf{.407_{\pm.227}}$
	Reasoning	800	$.417_{\pm.095}$	$\mathbf{.508_{\pm.094}}$	$.437_{\pm.104}$	$.430_{\pm.077}$	$.449_{\pm.113}$	$.465_{\pm.106}$
	Relations	565	$.411_{\pm.102}$	$\mathbf{.513_{\pm.110}}$	$.466_{\pm.090}$	$.495_{\pm.133}$	$.444_{\pm.089}$	$.500_{\pm.109}$
	Overall	1482	$.386_{\pm.057}$	$\mathbf{.484_{\pm.072}}$	$.415_{\pm.074}$	$.424_{\pm.067}$	$.426_{\pm.087}$	$.461_{\pm.086}$

Table 1: Compare the performance of agents with other multi-agent algorithms designed for multiplayer deduction games. SSA and FSA stand for Single Script Access and Full Script Access, respectively, representing the performance of agents when they have access to either only their own script or the scripts of all agents, without interacting with other agents.

Script	#Tokens	Stage	SSA	FSA	Agent's Response After Playing the Game			
					Werewolf	O-CoT	ThinkThrice	PLAYER*
Death Wears White *(9 players, 1 victim)*	3,190	Gameplay	–	–	3.797	4.643	3.992	3.363
		Evaluation	0.349	1.433	0.815	0.807	0.790	0.799
Ghost Revenge *(7 players, 3 victims)*	5,487	Gameplay	–	–	6.702	8.231	7.056	5.947
		Evaluation	0.418	1.945	1.006	1.004	0.995	1.012
Danshui Villa *(7 players, 2 victims)*	5,111	Gameplay	–	–	5.215	6.262	5.449	4.603
		Evaluation	0.351	1.415	0.834	0.816	0.833	0.825
Unfinished Love *(7 players, 2 victims)*	2,501	Gameplay	–	–	3.945	4.754	4.237	3.505
		Evaluation	0.230	0.872	0.487	0.502	0.499	0.494
Cruise Incident *(5 players, 1 victim)*	1,262	Gameplay	–	–	0.975	1.146	1.021	0.846
		Evaluation	0.064	0.327	0.162	0.165	0.164	0.163
Sin *(4 players, 1 victim)*	2,121	Gameplay	–	–	0.680	0.833	0.730	0.603
		Evaluation	0.061	0.287	0.141	0.140	0.141	0.142
Deadly Fountain *(4 players, 1 victim)*	1,852	Gameplay	–	–	0.671	0.810	0.724	0.596
		Evaluation	0.045	0.207	0.107	0.111	0.111	0.109
Unbelievable Incident *(5 players, 1 victim)*	3,182	Gameplay	–	–	1.304	1.567	1.367	1.150
		Evaluation	0.077	0.291	0.159	0.163	0.161	0.162
Desperate Sunshine *(4 players, 1 victim)*	3,370	Gameplay	–	–	0.803	0.972	0.847	0.701
		Evaluation	0.104	0.383	0.222	0.221	0.224	0.220
Riverside Inn *(4 players, 1 victim)*	1,909	Gameplay	–	–	0.633	0.762	0.682	0.561
		Evaluation	0.055	0.223	0.120	0.124	0.121	0.123
Solitary Boat Firefly *(6 players, 4 victims)*	8,893	Gameplay	–	–	7.799	9.257	8.252	6.800
		Evaluation	0.380	1.571	0.816	0.811	0.814	0.823
Manna *(6 players, 3 victims)*	9,028	Gameplay	–	–	5.805	6.925	6.108	5.028
		Evaluation	0.444	1.864	0.944	0.965	0.937	0.953
Overall	47,906	Gameplay	–	–	38.329	46.162	40.464	**33.702**
		Evaluation	2.578	10.819	5.813	5.831	5.789	5.825

Table 2: Compare the costs in US dollars($) of calling Openai API across multi-agent algorithms in MMGs setting, with Gameplay and Evaluation Stage. #*Tokens* represent the average length of each character's script. Costs are reported for one complete gameplay and one evaluation process for each script.

Efficiency and Cost Comparison As shown in Table 2, we delve into the efficiency and cost analysis across various methodologies implemented for agent interaction in MMGs, as detailed in our results table. The costs are presented in actual monetary values (US Dollars) associated with the use of Azure API, providing a direct measure of the computational expense incurred during both gameplay and evaluation stages [2].

Each script's complexity is indicated by the #*Tokens* column, reflecting the narrative depth and the number of characters, suspects, and victims involved. This complexity directly influences the API usage cost, as more intricate scenarios require more processing power for information handling. The costs are divided into Gameplay and Evaluation phases. During Gameplay, agents actively participate in the game, generating actions and responses. In the Evaluation phase, the performance of these agents is assessed based on the WellPlay Dataset. Notably, SSA and FSA methodologies do not incur costs during the Gameplay phase, as they were tested in a direct, non-interactive manner.

PLAYER* stands out for its cost-efficiency and performance. By employing the Action Space Refinement strategy, PLAYER* minimises unnecessary API calls, concentrating its investigative efforts on the most suspicious characters. This focus significantly reduces the computational resources required, thereby lowering the overall cost of operations. This strategic optimisation is evident across all scripts, with PLAYER* consistently registering lower costs in comparison to its counterparts during the Gameplay phase. Evaluation costs are similar across all methods due to a shared evaluation strategy, with variations in SSA and FSA costs arising from their differing levels of script access.

[2]Billing method details are available on the website `https://azure.microsoft.com/en-gb/pricing/details/cognitive-services/openai-service/`

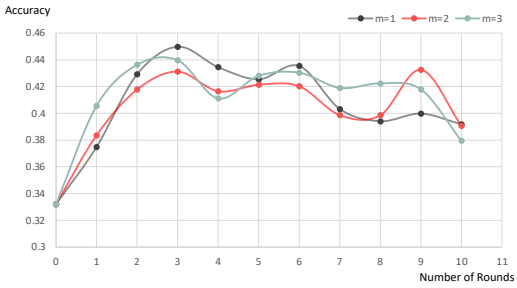

Figure 3: Comparison of agents' behaviour across different numbers of rounds, where each agent can ask a specific number of questions (denoted as m).

3.3 Ablation Studies

Performance peaks around round 3, after which it shows variability, with some rounds experiencing slight declines or plateauing in scores despite more rounds or questions. This indicates that after a rapid initial learning or adaptation phase, where agents effectively use additional questions to enhance their understanding and strategies, the value of information gained from conversations tends to converge. These results also provide empirical support for the assumption made in our methodology that the more inquiries we pose to an agent, the expected reward associated with questioning the same agent decreases. For the main results we reported, we use the original setting for number of rounds in MMG, which is 3, and based on the outcomes of the ablation studies, we chose the most effective number of questions to ask each round, which is 1 question.

4 Related Works

Multi-Agent Interaction Multi-agent reinforcement learning marking significant progress in complex games (Lanctot et al., 2017; Perolat et al., 2022; Bakhtin et al., 2023). However, these methods often require extensive time and computational resources and lack linguistic communication capabilities. With the emergence of LLMs, there's a shift of focus towards improving multi-agent language communication, evidenced by advancements in various games and scenarios, such as werewolf (Xu et al., 2023a), avalon (Wang et al., 2023; Shi et al., 2024), interactive narrative (Zhao et al., 2024b), MMGs (Wu et al., 2024), and survival games (Toy et al., 2024). Exemplified by AlphaGo (Silver et al., 2017), which demonstrated exceptional performance against human competitors, self-play learning frameworks (Fu et al., 2023; Chen et al., 2024) are proposed to improve LLMs' performance. This idea has also been adapted for evaluation through negotiation games (Davidson et al., 2024). Compared to classic methods (Wang & Shen, 2024), agents based on LLMs are capable of inferencing across a broader range of scenarios (Lin et al., 2023), even with some ability of theory of mind (Zhou et al., 2023) to infer other agent's mental states. However, they have also been found to inherit biases that limit their inferential abilities (Xie et al., 2023; Chuang et al., 2024). Works have also explored utilising LLMs as the environment (Zhang et al., 2024) or update actions (Zhao et al., 2024a) for agents.

Optimisation for Complicated Tasks Alignment through human feedback offers more consistent training compared to reinforcement learning (Liang et al., 2024), but obtaining this feedback can be expensive. To address this, approaches like self-instruct (Wang et al., 2022; Liu et al., 2023b), self-reflect (Yao et al., 2023), self-alignment (Sun et al., 2023; Li et al., 2024b), and few-shot planning (Song et al., 2023), have been introduced. These approach was also adapted to search for optimal tools (Du et al., 2024), interact with grounded environments (Ouyang & Li, 2023; Ismail et al., 2024). Adapting LLM-as-a-Judge prompting to evaluate performance (Yuan et al., 2024) or select agents (Liu et al., 2023c) has become a popular approach, but negative results on self-reflection were also investigated (Huang et al., 2024), leaving LLM's role as a self-reflective agent as an unresolved question. We were also inspired by stochastic search methods utilised for robots in planning optimal strategies

in complex environments (Gammell et al., 2020b; Liang et al., 2024), shares many similarities with optimisation tasks for agents (Singh et al., 2023).

5 Conclusion

PLAYER* addresses limitations of LLM-based agents in complex reasoning using an anytime sampling-based planner with sensors and pruners. Our approach enables efficient, question-driven searching. We propose a quantifiable evaluation method, contribute the WellPlay dataset, and demonstrate PLAYER*'s superiority, advancing effective reasoning agents in complex environments.

6 Ethics Statement

Please note that MMGs and our WellPlay dataset, created to assess agents' behaviors in MMGs, may include descriptions of violent events, actions, or characters. This content is included solely for academic, research, and narrative analysis purposes. It is not meant to glorify or trivialise violence in any form. We have informed annotators about the potential exposure to violent content. Users or researchers intending to use this dataset should be aware of this potential exposure and are advised to engage with the dataset in a professional and responsible manner. This dataset is unsuitable for minors and those sensitive to such content.

Acknowledgments

This work was supported in part by the UK Engineering and Physical Sciences Research Council through a New Horizons grant (EP/X019063/1) and a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2).

References

Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. In *the 11th International Conference on Learning Representations*. ICLR Press, 2023. URL https://openreview.net/forum?id=F61FwJTZhb.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *CoRR*, abs/2401.01335, 2024. URL http://arxiv.org/abs/2401.01335.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy T. Rogers. Simulating opinion dynamics with networks of llm-based agents. *CoRR*, 2024. URL https://arxiv.org/abs/2311.09618.

Tim R. Davidson, Veniamin Veselovsky, Martin Josifoski, Maxime Peyrard, Antoine Bosselut, Michal Kosinski, and Robert West. Evaluating language model agency through negotiations. *CoRR*, 2024. URL https://arxiv.org/abs/2401.04536.

Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of a*. *J. ACM*, 32(3):505–536, 1985. doi: 10.1145/3828.3830. URL https://doi.org/10.1145/3828.3830.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *CoRR*, 2024. URL https://arxiv.org/abs/2401.08281.

Yu Du, Fangyun Wei, and Hongyang Zhang. Anytool: Self-reflective, hierarchical agents for large-scale api calls. *CoRR*, abs/2402.04253, 2024. URL http://arxiv.org/abs/2402.04253.

Meta Fundamental AI Research Diplomacy Team FAIR, 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, Sasha Mitts, Adithya Renduchin-tala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378:1067–1074, 2022. URL https://www.science.org/doi/10.1126/science.ade9097.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from ai feedback. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://arxiv.org/abs/2305.10142.

Jonathan D. Gammell, Timothy D. Barfoot, and Siddhartha S. Srinivasa. Batch informed trees (bit*): Informed asymptotically optimal anytime search. *Int. J. Robotics Res.*, 39(5), 2020a. doi: 10.1177/0278364919890396. URL https://doi.org/10.1177/0278364919890396.

Jonathan D. Gammell, Timothy D. Barfoot, and Siddhartha S. Srinivasa. Batch informed trees (bit*): Informed asymptotically optimal anytime search. *Robotics Research*, 39, 2020b. URL https://dl.acm.org/doi/abs/10.1177/0278364919890396.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *the 12th International Conference on Learning Representations*. ICLR Press, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Seif Ismail, Antonio Arbues, Ryan Cotterell, René Zurbrügg, and Carmen Amo Alonso. Narrate: Versatile language architecture for optimal control in robotics. *CoRR*, 2024. URL https://arxiv.org/abs/2403.10762.

Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. Robotics Res.*, 30(7):846–894, 2011. doi: 10.1177/0278364911406761. URL https://doi.org/10.1177/0278364911406761.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3323fe11e9595c09af38fe67567a9394-Abstract.html.

Bibb Latané. The psychology of social impact. *American Psychologist*, 36:343–356, 1981. URL https://psycnet.apa.org/doiLanding?doi=10.1037%2F0003-066X.36.4.343.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. *CoRR*, abs/2303.17760, 2024a. URL http://arxiv.org/abs/2303.17760.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. In *the 12th International Conference on Learning Representations*. ICLR Press, 2024b. URL https://openreview.net/forum?id=pETSfWMUzy.

Jacky Liang, Fei Xia, Wenhao Yu, Andy Zeng, Montserrat Gonzalez Arenas, Maria Attarian, Maria Bauza, Matthew Bennice, Alex Bewley, Adil Dostmohamed, Chuyuan Kelly Fu, Nimrod Gileadi, Marissa Giustina, Keerthana Gopalakrishnan, Leonard Hasenclever, Jan Humplik, Jasmine Hsu, Nikhil Joshi, Ben Jyenis, Chase Kew, Sean Kirmani, Tsang-Wei Edward Lee, Kuang-Huei Lee, Assaf Hurwitz Michaely, Joss Moore, Ken Oslund, Dushyant Rao, Allen Ren, Baruch Tabanpour, Quan Vuong, Ayzaan Wahid, Ted Xiao, Ying Xu, Vincent Zhuang, Peng Xu, Erik Frey, Ken Caluwaerts, Tingnan Zhang, Brian Ichter, Jonathan Tompson, Leila Takayama, Vincent Vanhoucke, Izhak Shafran, Maja Mataric, Dorsa Sadigh, Nicolas Heess, Kanishka Rao, Nik Stewart, Jie Tan, and Carolina Parada.

Learning to learn faster from human feedback with language model predictive control. *CoRR*, 2024. URL https://arxiv.org/abs/2402.11450.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://openreview.net/forum?id=Rzk3GP1HN7.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172, 2023a. doi: 10.48550/ARXIV.2307.03172. URL https://doi.org/10.48550/arXiv.2307.03172.

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. Agentbench: Evaluating llms as agents. *CoRR*, 2023b. URL https://arxiv.org/abs/2308.03688.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *CoRR*, 2023c. URL https://arxiv.org/abs/2310.02170.

Siqi Ouyang and Lei Li. Autoplan: Automatic planning of interactive decision-making tasks with large language models. In *Proceedings of the Findings of the Association for Computational Linguistics*, pp. 3114–3128. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-emnlp.205/.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. Association for Computing Machinery, 2023. URL https://openreview.net/pdf?id=p40XRfBX96.

Julien Perolat, Bart de Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 10, 2022. URL https://www.science.org/doi/10.1126/science.add4679.

Zijing Shi, Meng Fang, Shunfeng Zheng, Shilong Deng, Ling Chen, and Yali Du. Cooperation on the fly: Exploring language agents for ad hoc teamwork in the avalon game. *CoRR*, abs/2312.17515, 2024. URL http://arxiv.org/abs/2312.17515.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of stratego with model-free multiagent reinforcement learning. *Nature*, 550:354–359, 2017. URL https://www.nature.com/articles/nature24270.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: program generation for situated robot task planning using large language models. *Autonomous Robots*, 47:999–1012, 2023. URL https://link.springer.com/article/10.1007/s10514-023-10135-3.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M. Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Institute

of Electrical and Electronics Engineers Inc., 2023. URL https://openaccess.thecvf.com/content/ICCV2023/papers/Song_LLM-Planner_Few-Shot_Grounded_Planning_for_Embodied_Agents_with_Large_Language_ICCV_2023_paper.pdf.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://openreview.net/pdf?id=p40XRfBX96.

Jason Toy, Josh MacAdam, and Phil Tabor. Metacognition is all you need? using introspection in generative agents to improve goal-directed behavior. *CoRR*, abs/2401.10910, 2024. URL http://arxiv.org/abs/2401.10910.

Yaacov Trope and Nira Liberman. Construal-level theory of psychological distance. *Psychol Rev*, 117:440–463, 2010. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3152826/.

Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. Avalon's game of thoughts: Battle against deception through recursive contemplation. *CoRR*, abs/2310.01320, 2023. URL http://arxiv.org/abs/2310.01320.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. In *the 61st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2023.acl-long.754.pdf.

Zongshun Wang and Yuping Shen. Bilateral gradual semantics for weighted argumentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. MIT Press, 2024. URL https://ojs.aaai.org/index.php/AAAI/article/view/28945.

Dekun Wu, Haochen Shi, Zhiyuan Sun, and Bang Liu. Deciphering digital detectives: Understanding llm behaviors and capabilities in multi-agent mystery games. *CoRR*, abs/2312.00746, 2024. URL http://arxiv.org/abs/2312.00746.

Zhuohan Xie, Trevor Cohn, and Jey Han Lau. The next chapter: A study of large language models in storytelling. In *Proceedings of the 16th International Natural Language Generation Conference*, pp. 323–351. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.inlg-main.23/.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *CoRR*, abs/2309.04658, 2023a. URL http://arxiv.org/abs/2309.04658.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023b. URL https://arxiv.org/pdf/2309.04658.pdf.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *CoRR*, abs/2401.10020, 2024. URL http://arxiv.org/abs/2401.10020.

Alex Zhang, Khanh Nguyen, Jens Tuyls, Albert Lin, and Karthik Narasimhan. Language-guided world models: A model-based approach to ai control. *CoRR*, 2024. URL https://arxiv.org/abs/2402.01695.

Haiteng Zhao, Chang Ma, Guoyin Wang, Jing Su, Lingpeng Kong, Jingjing Xu, Zhi-Hong Deng, and Hongxia Yang. Empowering large language model agents through action learning. *CoRR*, 2024a. URL https://arxiv.org/abs/2402.15809.

Runcong Zhao, Wenjia Zhang, Jiazheng Li, Lixing Zhu, Yanran Li, Yulan He, and Lin Gui. Narrativeplay: An automated system for crafting visual worlds in novels for role-playing. In *the 38th Annual AAAI Conference on Artificial Intelligence*. MIT Press, 2024b. URL https://arxiv.org/abs/2310.01459.

Runcong Zhao, Qinglin Zhu, Hainiu Xu, Jiazheng Li, Yuxiang Zhou, Yulan He, and Lin Gui. Large language models fall short: Understanding complex relationships in detective narratives. *CoRR*, abs/2402.11051, 2024c. URL http://arxiv.org/abs/2402.11051.

Yihan Zhao, Rong Chen, Mitsuyasu Yabe, Buxin Han, and Pingping Liu. I am better than others: Waste management policies and self-enhancement bias. *Sustainability*, 13, 2021. URL https://www.mdpi.com/2071-1050/13/23/13257.

Pei Zhou, Aman Madaan, Srividya Pranavi Potharaju, Aditya Gupta, Kevin R. McKee, Ari Holtzman, Jay Pujara, Xiang Ren, Swaroop Mishra, Aida Nematzadeh, Shyam Upadhyay, and Manaal Faruqui. How far are large language models from agents with theory-of-mind? *CoRR*, 2023. URL https://arxiv.org/abs/2310.03051.

A MMGs Rules and Procedure

A.1 Detailed Rules

Rule 1: The total number of players participating in the game depends on the script. There may be one or more players who are the murderer(s), while the rest are civilians.

Rule 2: The goal of the game is for civilian players to collaborate and face a meticulously planned murder case together, collecting evidence and reasoning to identify the real murderer among the suspects, all the while ensuring they are not mistaken for the murderer; murderer players must concoct lies to hide their identity and avoid detection, while also achieving other objectives in the game.

Rule 3: Throughout the game, only murderer players are allowed to lie. To conceal their identity, murderers may choose to frame others to absolve themselves of guilt; non-murderer players (civilians) must answer questions from other players and the host honestly and provide as much information as they know about the case to help uncover the truth.

Rule 4: At the start of the game, each player receives their character script from the host, which contains information about their role and identity.

Rule 5: Other players cannot see the content of each player's character script, so players must and can only collect information about other players through interaction after the game starts.

Rule 6: In the voting phase, each player needs to cast their vote for who they think is the murderer in each case. If the player with the most votes is the murderer, the civilian players win. Otherwise, the murderer players win.

A.2 Procedure

Stage 1: Distribution of Character Scripts The host distributes character scripts to each player. These scripts contain the player's name, role (murderer or civilian), and a brief character backstory.

Stage 2: Self-Introduction Session Players introduce their characters to the group, laying the groundwork for the game's interactions.

Stage 3: Rounds of Open Questioning The game progresses through three rounds of open questioning. Players take turns to ask and answer questions, aiming to gather information about others.

Stage 4: Voting In this stage, players vote anonymously to determine their suspicious regarding the identity of the murderer. Each player has one vote.

Stage 5: Outcome Reveal The game concludes with the announcement of the voting results, revealing whether the civilian players successfully identified the murderer or not.

A.3 Example: Solitary Boat Firefly Script

As an illustrative example, we examine the *Solitary Boat Firefly* murder mystery script, involving six players: ["Tian Chou", "Zhou Lianyi", "Xi Yan", "Yu Sunian", Yannan", and "Zhou Chitong"], along with four victims: ["Zhou Mengdang", "Bao Liu", "Cui Shouheng", and "Wang Xi Rong"]. We model these characters through a set of agents, denoted as $\mathcal{A} = \{a_i\}_{i=1}^{N_a}$, where $N_a = 6$ corresponds to the number of players. In parallel with the four victims, we define a set of victims $\mathcal{V} = \{v_i\}_{i=1}^{N_v}$, where $N_v = 4$, representing the total number of victims in the scenario.

Taking a closer look at "Tian Chou", depicted as the murderer player a_1, responsible for the demise of two among the four victims, her characterization unfolds as follows:

- **Character Background (C_1)**: "Originating from the Sun lineage, you are endearingly called 'Tian'er'. Your birth year was the twelfth of Guangxu's reign during the Qing Dynasty (1886), marking the beginning of a life filled with extraordinary episodes..."
- **Suspicion State (s_1)**: The script features four victims, each harboring suspicions towards the remaining five players. Consequently, s_1 is represented as a 4×5 matrix, with each row corresponding to a victim and each column reflecting the suspicions they hold against other players. For example, a first row of $[1, 0, 0, 1, 0]$ signifies that victim "Cui Shouheng" suspects both "Zhou Lianyi" and "Yannan". So we can get *suspect_list* of victim "Cui Shouheng" is ["Zhou Lianyi" ,"Yannan"]
- **Individual Objectives (o_i)**: The personal objectives for Tian Chou include:
  1. Concealing that you killed "Zhou Mengdang".
  2. Concealing that you killed "Bao Liu".
  3. Find out the truth about "Taitai's death".
  4. Conceal your relationship with "Zhou Chitong".
  5. ...

During the game, the dialogue history is recorded in D, and as the game progresses, inquiries and responses are conducted in accordance with Algorithm 1, leading to updates in s_i.

B The WellPlay Dataset

B.1 Details of Dataset

In this work, we leverage the *Conan* dataset, originally constructed by Zhao et al. (2024c), which comprises scripts from MMGs, including detailed annotations of character relationships. Our bilingual dataset, encompassing both Chinese and English versions, is derived from this foundational work. Our dataset consists of two main components: scripts and evaluation questions.

Each script features distinct narratives for individual characters. We have rephrased the original script from the *Conan* dataset into two parts:

Script	Agents	Victims	#token(CN)		#token(EN)		Question			
			avg	overall	avg	overall	Objective	Reasoning	Relations	overall
Death Wears White	9	1	3190.67	28716	1742.33	15681	10	102	72	184
Ghost Revenge	7	3	5487.86	38415	3960.43	27723	19	152	69	240
Danshui Villa	7	2	5111.29	35779	3338.57	23370	12	128	63	203
Unfinished Love	7	2	2501.00	17507	1651.71	11562	12	61	72	145
Cruise Incident	5	1	1262.60	6313	808.00	4040	4	24	30	58
Sin	4	1	2121.25	8485	1378.00	5512	3	20	21	44
Deadly Fountain	4	1	1852.50	7410	1193.75	4775	3	21	12	36
Unbelievable Incident	5	1	3182.40	15912	2012.40	10062	4	24	15	43
Desperate Sunshine	4	1	3370.25	13481	2218.50	8874	3	18	36	57
Riverside Inn	4	1	1909.50	7638	1257.00	5028	3	18	18	39
Solitary Boat Firefly	6	4	8893.67	53362	6874.00	41244	20	109	69	198
Manna	6	3	9028.17	54169	6492.33	38954	24	123	88	235
Avg	5.67	1.75	3992.60	23932.25	2743.92	16402.08	9.75	66.67	47.08	125.50
Sum	68	21	47911.16	287187	32927.02	196825	117	800	565	1482

Table A1: Dataset Statistics. *Agents* is the count of players, *Victims* is the number of script victims, #*token(CN)* and #*token(EN)* are the token counts in the Chinese and English dataset versions, respectively. *Avg* shows the average script length per character, *Overall* is the total script token count, and *Question* enumerates the number of questions by types.

1. **Background Script**. For each character, there is corresponding background script includes all the information from their perspective. For example, for *"Sylvia Costa"* in the script *"Death Wears White"*, it is:

   You are the head nurse of the emergency ward.You climbed to this position for your hard work and were proud. You are a very professional person and highly appreciated by colleagues. This job is perfect for you, except for a problem - you have not made enough money. Your salary is actually not enough for you to live a decent life, far from paying enough to take care of Mother's overhead ...

2. **Personal Objectives**. For each character, there are corresponding objectives that guide their actions. For example, for *"Sylvia Costa"* in the script *"Death Wears White"*, it is:

   1. Ensure that no one will discover your illegal organ trafficking activities;
   2. Ensure that the kidnappers leave without being injured or killing anyone - you want to ensure that the police do not investigate deeply enough to discover your organ trafficking. The further away you are from the police, the safer you feel;
   3. ...

The evaluation questions are devised based on our annotations and aim to assess the understanding of the intricate relationships and narratives present within the game scripts. These questions are segmented into three categories, with each category of questions and examples shown in Table A2.

1. **Objective:** This category includes common objectives shared among agents, such as finding the perpetrator(s).
2. **Reasoning:** Comprising six types of questions, this section tests the models' reasoning capabilities across various aspects:
   - **How:** How the murder was committed.
   - **Why:** The motive behind the murder.
   - **Relationship:** The relationship between the murderer and the victim.
   - **Where:** The location where the murder took place.
   - **When:** The time at which the murder occurred.
   - **Suspect:** Identify the two most suspicious individuals. Analogous to human reasoning, it may be challenging to definitively pinpoint the suspect, yet it is possible to determine those who seem most suspect.
3. **Relations:** This segment examines the model's capacity to comprehend complex narratives by querying about the relationships between characters. Utilizing the

Type	Aspects	Examples (The correct answer has been highlighted in bold.)
A (score 10)	Who	Who killed Hans Li Morette? **A. Gale Li Morette** B.Nurse head [Sylvia Costa] C. Drake Li Morette D. Frank Bijeli
B (score 5)	How	How did Hans Li Morette die? **A. Shot to death** B. Beaten to death C. Poisoned to death by poison D.Drowned by water
	Why	What was the motive behind the killer killing Hans Li Morette? A. Love killing B. Vendetta **C. Interest** D. Accidental killing
	Relationship	What is the relationship between Murderer and Victim Hans Li Morette? A. Enemies B. Colleague C. Friend **D. Wife**
	Where	Where was Hans Li Morette killed? A. Emergency room B. Johnson's House **C. Laboratory** D. Dressing room
	When	When was Hans Li Morette killed? A. This afternoon from 5:00 to 5:30 **B. This afternoon from 6:30 to 7:00** C. Tonight from 7:00 to 7:30 D. This morning from 6:30 to 7:00
	Suspect	Please select the two people you most suspect of killing Hans Li Morette **A. Gale Li Morette** B. Nurse head [Sylvia Costa] C. Drake Li Morette D. Frank Bijeli
C (score 2)	Three relationships	What is the non-existent relationship between Hans Li Morette and Andrew Paloski? A. Andrew Paloski is colleague of Hans Li Morette, B. Andrew Paloski is mentor of Hans Li Morette, C. Andrew Paloski is jealous of Hans Li Morette, **D. Hans Li Morette is future daughter in law of Andrew Paloski**
	Two relationships	What is the relationship between Father Tom and Tony? **A. Tony is manipulated by x and deceived by x of Father Tom** B. Father Tom is authority over x and student of Tony C. Father Tom is student and ex-girlfriend of Tony D. Father Tom is ex-girlfriend and admired by x of Tony
	One relationships	What is the relationship between Father Tom and Drake Li Morette? **A. Drake Li Morette is doctor of Father Tom** B. Father Tom is helped by Drake Li Morette C. Father Tom is step-brother of Drake Li Morette D. Father Tom is hate of Drake Li Morette

Table A2: Examples of Each Type of Our Evaluation Questions

Index	Question
1	What was your timeline on the day of the incident?
2	How would you describe your relationship with the victim?
3	When was the last time you saw the victim?
4	Do you know if the victim had any enemies or conflicts with anyone?
5	What details or anomalies did you notice at the scene of the crime?
6	Did the victim mention anything to you or others that made them worried or fearful recently?
7	Did you notice any unusual people or behaviors on the day of the incident?
8	How much do you know about the victim's secrets or personal life?
9	Were there any items or remains found at the crime scene that could be related to the crime?
10	Do you have any personal opinions or theories about the case?

Table A3: Predefined questions for Werewolf method.

relationship annotations from the *Conan* dataset(Zhao et al., 2024c), we approach

this analysis through various questioning strategies based on the number of relationships between two characters:

- For three relationships, we employ the elimination method, asking the model to identify the incorrect relation among the given options.
- For two relationships, we connect the correct two relationships with “and" and introduce a distractor from other relationship categories.
- For single relationships, we list the correct option alongside distractors randomly selected from other categories.

For each individual mentioned in the purpose section, we extract three significant relationships associated with them. In this context, “significant" is defined as the characters who have the most complex connections with the individual, that is, the top three characters who share the greatest variety of relationships with them. For instance, in the provided example, Andrew Paloski is Hans Li Morette’s colleague, mentor, and is also jealous of him.

For various types of questions, we assign different weights based on the original scoring system of the script. Specifically, Type A questions are valued at 10 points, Type B questions at 5 points, and Type C questions at 2 points.

Detailed script statistics and evaluation question metrics are presented in TableA1.

B.2 Overall Performance Computing

In calculating the overall score for performance, we have employed both the weighted mean and the weighted standard deviation. The weighted mean is computed by considering the count of questions for a specific category across various scripts as the weight. For the overall score, the total possible score for each script serves as the weight. This method allows us to adjust the influence of each category and script based on its significance and scale, thus providing a more nuanced and accurate reflection of performance.

The weighted mean is calculated as:

$$\bar{x}_w = \frac{\sum_{i=1}^{n}(w_i \cdot x_i)}{\sum_{i=1}^{n} w_i}$$

The weighted standard deviation, which measures the spread of the scores, is calculated using the weighted variance:

$$s_w^2 = \frac{\sum_{i=1}^{n} w_i \cdot (x_i - \bar{x}_w)^2}{\sum_{i=1}^{n} w_i}$$

And the weighted standard deviation is the square root of the weighted variance:

$$s_w = \sqrt{s_w^2}$$

C Implementation

C.1 Implementation Details

RAG (Retrieval-Augmented Generation) For retrieval enhancement, we utilised the FAISS[3] library to build a vector database, creating FAISS indices using the L2 distance. Embeddings were obtained via the Azure API’s `text-embedding-ada-002` service. Scripts were stored in segments, with each segment having a maximum length of 50 tokens. For dialog records, a question-and-answer pair was stored as a single segment. During retrieval, the maximum script length and dialog length inserted into the prompt were both set to 4000 tokens. For evaluation, these maximum lengths were increased to 5000 tokens.

[3]https://github.com/facebookresearch/faiss

Experiment Following the results of our ablation studies, the gameplay phase was structured to ask one question per round over three rounds. After the game concluded, the evaluation phase consisted of three separate evaluations, with the final results being the average of these evaluations.

C.2 Comparison Models

In this section, we provide a detailed overview of the methodologies compared in our study. Given the game's rules and questioning sequence in our comparisons, we segment the discussion into two phases: the questioning phase and the answering phase.

Considering the strategic core of the game, a direct confession from the murderer would compromise its competitive nature. Hence, in all the methodologies we explore, we maintain that the murderer's responses follow a predefined template, detailed in C.6.2. This method preserves the game's integrity and level of challenge.

1. **Werewolf**(Xu et al., 2023a)
   Question Generation:
   a. Selecting questions from a predefined question written by the human specialist, based on the current dialogue and script. The list of expertly devised questions is presented in Table A3.
   b. Formulating questions by the selected predefined questions and the ongoing dialogue and script.
   Answering Questions:
   a. Responding based on the current script and dialog history.
   b. Reflecting on the initial response in light of the dialog history.
   c. Generating the final answer after reflection.
2. **Objective-Guided Chain of Thought (O-CoT)**(Park et al., 2023; Zhao et al., 2024b)
   Question Generation: Entails two critical steps:
   a. Sequentially reflecting on whether current objectives have been met, with considerations spanning multiple goals such as identifying the murderer, uncovering hidden relationships, or concealing facts. For example, Who is the murderer?
   b. Crafting questions based on these reflections and the current narrative and dialogue.
   Answering Questions:
   a. Answers are formulated leveraging the narrative and dialog history.
3. **ThinkThrice**(Wu et al., 2024)
   Question Generation:
   a. Questions are generated based on the script and dialog history.
   Answering Questions:
   a. Extracting timelines relevant to the victim.
   b. Evaluating each timeline's relevance to answering the question.
   c. Responding based on the dialog history, character relationships, and the relative script.
   d. Ensuring timelines that aid in answering the question are included in the response. If these are initially missed, the model will later augment and clarify the answer with the required timelines.
4. **The SSA (Single Script Access)**
   The SSA assesses agents' performance when limited solely to their scripts, serving as a baseline for initial search efforts.
5. **The FSA (Full Script Access)**
   The FSA evaluates performance under conditions of unrestricted access to all scripts, representing an ideal search endpoint.

C.3 Performance Evaluation Results for English Dataset

Due to budgetary constraints, we only evaluated four scripts from the English dataset, with the performance results reported in Table A4. For the same scripts, we found that the experiments conducted on the English corpus corroborate the results obtained from the Chinese corpus. Additionally, we observed that the performance of agents based on the English corpus surpassed those based on the Chinese corpus, showing the differential inferencing abilities of LLMs across languages. This discrepancy could be attributed to language biases inherent in the training data utilised for these models.

Script	Evaluation	#QA	SSA	FSA	Agent's Response After Playing the Game			
					Werewolf	O-CoT	ThinkThrice	PLAYER*
Death Wears White (9 players, 1 victim)	Objective	10	$.200_{\pm.173}$	$.900_{\pm.100}$	$.300_{\pm.100}$	$.267_{\pm.058}$	$.300_{\pm.000}$	$.267_{\pm.058}$
	Reasoning	102	$.350_{\pm.044}$	$.520_{\pm.026}$	$.356_{\pm.037}$	$.363_{\pm.049}$	$.399_{\pm.006}$	$.441_{\pm.030}$
	Relations	72	$.547_{\pm.008}$	$.445_{\pm.026}$	$.495_{\pm.016}$	$.495_{\pm.057}$	$.398_{\pm.032}$	$.491_{\pm.021}$
	Overall	184	$.367_{\pm.040}$	$.549_{\pm.012}$	$.375_{\pm.033}$	$.375_{\pm.026}$	$.385_{\pm.010}$	$.427_{\pm.029}$
Ghost Revenge (7 players, 3 victims)	Objective	19	$.403_{\pm.061}$	$.561_{\pm.132}$	$.439_{\pm.109}$	$.211_{\pm.106}$	$.211_{\pm.106}$	$.526_{\pm.106}$
	Reasoning	152	$.456_{\pm.004}$	$.507_{\pm.007}$	$.439_{\pm.031}$	$.539_{\pm.023}$	$.441_{\pm.017}$	$.423_{\pm.031}$
	Relations	69	$.309_{\pm.009}$	$.261_{\pm.025}$	$.328_{\pm.059}$	$.304_{\pm.015}$	$.280_{\pm.017}$	$.353_{\pm.059}$
	Overall	240	$.428_{\pm.010}$	$.485_{\pm.024}$	$.425_{\pm.002}$	$.452_{\pm.023}$	$.380_{\pm.010}$	$.432_{\pm.013}$
Danshui Villa (7 players, 2 victims)	Objective	12	$.167_{\pm.083}$	$.278_{\pm.048}$	$.472_{\pm.048}$	$.333_{\pm.000}$	$.305_{\pm.048}$	$.389_{\pm.096}$
	Reasoning	128	$.325_{\pm.031}$	$.427_{\pm.009}$	$.401_{\pm.020}$	$.373_{\pm.020}$	$.344_{\pm.014}$	$.357_{\pm.033}$
	Relations	63	$.328_{\pm.056}$	$.460_{\pm.028}$	$.391_{\pm.066}$	$.312_{\pm.106}$	$.376_{\pm.046}$	$.407_{\pm.075}$
	Overall	203	$.292_{\pm.012}$	$.412_{\pm.017}$	$.409_{\pm.009}$	$.359_{\pm.010}$	$.343_{\pm.010}$	$.369_{\pm.033}$
Unfinished Love (7 players, 2 victims)	Objective	12	$.167_{\pm.000}$	$.195_{\pm.048}$	$.389_{\pm.048}$	$.417_{\pm.000}$	$.361_{\pm.048}$	$.528_{\pm.127}$
	Reasoning	61	$.536_{\pm.018}$	$.557_{\pm.028}$	$.650_{\pm.038}$	$.612_{\pm.009}$	$.634_{\pm.038}$	$.656_{\pm.000}$
	Relations	72	$.491_{\pm.016}$	$.551_{\pm.016}$	$.509_{\pm.032}$	$.481_{\pm.021}$	$.592_{\pm.008}$	$.500_{\pm.037}$
	Overall	145	$.446_{\pm.006}$	$.479_{\pm.029}$	$.560_{\pm.011}$	$.538_{\pm.010}$	$.566_{\pm.009}$	$.589_{\pm.018}$

Table A4: Compare the performance of agents with other multi-agent algorithms designed for multiplayer deduction games. SSA and FSA stand for Single Script Access and Full Script Access, respectively, representing the performance of agents when they have access to either only their own script or the scripts of all agents, without interacting with other agents.

C.4 Experiments with Open-Source LLMs

Models	Llama2 70b	Llama2 13b	Llama2 7b	gemma 7b
Overall Score	0.312	0.281	0.267	0.273

Table A5: Compare the performance of PLAYER* method with different open-source LLMS.

In addition to experimenting with GPT-3.5-turbo-16k 0613, we also explored Llama2 (70b, 13b, 7b) and Gemma 7b. These models were tested using the scenario "Solitary Boat Firefly", and the overall results are reported in Table A5. The findings indicate that all four models scored significantly lower than GPT-3.5, presumably due to the limitations imposed by a 4k context window. This limitation likely hindered the models' ability to encapsulate sufficient relevant information within such a constrained window. After thorough testing, we observed that despite being evaluated on a Chinese dataset, Llama2 70b primarily resorted to English conversations due to its limited proficiency in Chinese. It responded in English even to Chinese prompts. Other models struggled even more with executing the prompts as required. This limitation greatly hindered their performance in complex gaming situations, such as MMGs. Therefore, we decided to exclusively use GPT-3.5 for future experiments, given its ability to navigate these intricate scenarios.

C.5 Sensors

This section provides a detailed explanation of the sensors employed in the 2.3, which are essential for both the Search by Questioning and Action Space Refinement components.

- **Emotion Sensor:** Assesses emotional inclination towards a character. Used in both Search by Questioning and Action Space Refinement. It categorises emotional inclination into “Positive", “Neutral", or “Negative".
- **Motivation Sensor:** Evaluates the character’s relationship with the victim and the presence of a motive for the crime. It is active in both phases, with choices being “Yes" or “No".
- **Suspicion Sensor:** Determines if a character is a suspect by analysing their opportunity to commit the crime. It applies to both stages, with responses “Yes" or “No".
- **Information Value Sensor:** Exclusive to Action Space Refinement, it estimates the probability of obtaining valuable information from further questioning. The choices are “High", “Medium", or “Low".

```
{
  {
    "name": "emotion",
    "for_Search_by_Questioning": True,
    "for_Action_Space_Refinement": True,
    "sensor_prompt": "What is your emotional inclination towards the
    character mentioned above?",
    "choices": ["Positive", "Natural", "Negative"],
  }
  {
    "name": "motivation",
    "for_Search_by_Questioning": True,
    "for_Action_Space_Refinement": True,
    "sensor_prompt": "What do you think is the relationship between the
    character mentioned above and the victim? \n  Do you think the
    character mentioned above has a motive for the crime?",
    "choices": ["Yes", "No"],
  }
  {
    "name": "suspicion",
    "for_Search_by_Questioning": True,
    "for_Action_Space_Refinement": True,
    "sensor_prompt": "Do you think the character mentioned above is a
    suspect? \n  This refers to whether the character objectively had the
     opportunity to commit the crime, such as if someone saw the
    character at the scene of the crime.",
    "choices": ["Yes", "No"],
  }
  {
   "name": "information value",
    "for_Search_by_Questioning": False,
    "for_Action_Space_Refinement": True,
    "sensor_prompt": "What do you think is the probability of obtaining
    valuable information by continuing to question the character
    mentioned above?",
    "choices": ["High", "Medium", "Low"],
  }
}
```

C.6 Prompt

C.6.1 System Prompt

System Prompt designed to introduce the gameplay of the MMG, along with providing essential information about the agents involved. The prompts dynamically adapt to include {character_name}, representing the agent’s character in the game, and {character_name_list}, listing the characters played by other participants.

For Civilian Players:

```
You are playing a game called "Murder Mystery" with other players, which
is based on textual interaction. Here are the game rules:

Rule 1: The total number of players participating in the game depends on
the script. There may be one or more players who are the murderer(s),
while the rest are civilians.
Rule 2: The goal of the game is for civilian players to collaborate and
face a meticulously planned murder case together, collecting evidence and
 reasoning to identify the real murderer among the suspects, all the
while ensuring they are not mistaken for the murderer; murderer players
must concoct lies to hide their identity and avoid detection, while also
achieving other objectives in the game.
Rule 3 Throughout the game, only murderer players are allowed to lie. To
conceal their identity, murderers may choose to frame others to absolve
themselves of guilt; non-murderer players (civilians) must answer
questions from other players and the host honestly and provide as much
information as they know about the case to help uncover the truth.
Rule 4: At the start of the game, each player receives their character
script from the host, which contains information about their role and
identity.
Rule 5: Other players cannot see the content of each player's character
script, so players must and can only collect information about other
players through interaction after the game starts.
Rule 6: In the voting phase, each player needs to cast their vote for who
 they think is the murderer in each case. If the player with the most
votes is the murderer, the civilian players win. Otherwise, the murderer
players win.


Gameplay:
The game has one or more acts. At the beginning of the game, players
introduce themselves according to the script, and in each act, you will
receive more plot information. In each act, you can ask questions, share
your observations, or make deductions to help solve the murder case.
The goal is to identify the true murderer and explain their motive. If
you are the true murderer, you must hide your identity and avoid
detection.

Now, you are playing the role of {character_name}, and the other players
are {character_name_list}.
You are not the murderer. Please collaborate with the other civilian
players to achieve your personal objective while finding the true culprit
!
```

For Killer Players, we replace the last paragraph with:

```
  You are the murderer who kills {victims}, and you haven't killed anyone
   else. Please hide the fact that you committed the murder by
  fabricating lies and other information, and accomplish your personal
  objective!
```

C.6.2 Gameplay Prompt

Self-Introduction Prompt designed to facilitate introductions based on the agent's script and objectives. Within this prompt, {current_script} represents the character's script, and {goal} represents the objectives of the character within that script.

For Civilian Players:

```
Your Script is {current_script}.
Your goal is {goal}.
You are not a murderer, so tell more details to help find the murderer.
```

```
If you have something to hide, then be sure not to divulge the relevant
information! Don't reveal your goals.
Please introduce yourself.
```

For Killer Players:

```
Your Script is {current_script}.
Your goal is {goal}.
If you have something to hide, then be sure not to divulge the relevant
information!
Please introduce yourself.
You are a murderer, so You can lie to cover yourself!
```

Sensor Prompt As outlined in sections 2.3 and C.5, the Sensor Prompt is crafted to collect a wide array of crucial information. Within this framework, the placeholders {victim} and {character} are employed to reference the victim's name and the agent being questioned, respectively. Meanwhile, {current_script} and {dialog_history} retrieve the script and dialog logs relevant to both the deceased and the agent under scrutiny, utilising the RAG technique. The constructs {sensor_Prompt} and {choices} have been formulated in C.5 within the Sensor Prompt segment.

For Civilian Players:

```
{victim} was murdered, you are not the murderer, you need to try to find
the murderer.
Your Script is about {character} is {current_script}.
The dialog history about {character} is {dialog_history}.
Be warned, if it's a murderer's word it might deceive you.
Based on the information above, {sensor}
Please answer {choices} and explain your reasoning in one or two
sentences.
```

For Killer Players:

```
{victim} was murdered, you are the murderer, but you need to hide
yourself, and pretend you're not the murderer.
Your Script is about {character} is {current_script}.
The dialog history about {character} is {dialog_history}.
Based on the information above, {sensor}
Please answer {choices} and explain your reasoning in one or two
sentences.
```

Search by Questioning Prompt As outlined in sections 2.3, the Search by Questioning Prompt is developed based on data acquired from sensors. It includes variables like {victim}, {character}, {current_script}, {dialog_history}. Additionally, it integrates a summary, {summary}, synthesized from the sensor-collected data. The element {question_number} denotes the total questions permitted, with a detailed discussion on the optimal number of questions presented in the 3.3 chapter.

For Civilian Players:

```
{victim} was murdered, you are not the murderer, you need to try to find
the murderer.
Your Script is about {character} is {current_script}.
The dialog history about {character} is {dialog_history}.
{summary}
You can ask {character} {question_number} questions. What would you ask?
Please include the victim's name in your question when asking,
Since the murderer will lie, you can ask questions based on the loopholes
 and contradictions in what they have previously said.
Please respond in the JSON format without any additional comments.
For example,
```

```
{
  'Question1': 'Your question',
  'Question2': 'Your question'
}
```

For Killer Players:

```
{victim} was murdered, you are the murderer, But you need to hide
yourself, pretend you're not a murderer, and ask questions of other
people pretending you suspect the other person is a murderer.
Your Script is about {character} is {current_script}.
The dialog history about {character} is {dialog_history}.
{summary}
You can ask {character} {question_number} questions. What would you ask?
Please include the victim's name in your question when asking.
Please respond in the JSON format without any additional comments.
For example,
{
  'Question1': 'Your question',
  'Question2': 'Your question'
}
```

Action Space Refinement Prompt As outlined in sections 2.3, the Action Space Refinement prompt serves as a strategic tool for narrowing down the suspect list, effectively reducing the search domain to augment the efficiency and performance of the algorithm. Within this setup, the term {victim} refers to the individual who has been harmed, {summary} synthesised from the sensor-collected data, and {character_suspect} identifies the roster of individuals under suspicion. Initially, this roster includes all participating agents, excluding the itself.

```
{victim} was murdered.
You think {character_suspect} are suspected of killing {victim}, and your
 reasons for suspecting them are respectively:
{summary}
Please select several people you think are the most suspicious. You can
choose one or more, Please try to reduce the number of suspects.
Please respond in the JSON format without any additional comments.
For example,
{
  'suspicion': ["character_name1", "character_name2"]
}
```

Question Reply Prompt The Question Reply Prompt is designed for responding to inquiries, where {character} denotes the name of the character asking you a question, and {question} is the question itself. {current_script} refers to the script extracted using RAG that pertains to both the deceased and the agent being questioned; {dialog_history} is the dialog log related to the deceased and the interrogated agent, also extracted using RAG.

For Civilian Players:

```
{character} ask you a question: {question}
Your Script relative to the question is {current_script}.
The dialog history relative to the question is {dialog_history}.
What you need to pay attention to is {goal}.
Be warned, in the dialog history, if it's a murderer's word it might
deceive you.
Please answer the question: "{question}" based on the information above.
You are not the murderer, and you need to work hard to find the murderer.
 Therefore, provide as much information as possible, such as clues
related to the timeline, emotional information, etc.
Please answer the questions from a first-person perspective, rather than
saying what someone else said.
```

For Killer Players:

```
{character} ask you a question : {question}
Your Script relative to the question is {current_script}.
The dialog history relative to the question is {dialog_history}.
What you need to pay attention to is {goal}.
Be warned, in the dialog history, if it's a murderer's word it might
deceive you.
Please answer the questions: "{question}" based on the information above.
You are the murderer. Please hide the fact that you killed {victim}. You
can fabricate lies.
Please answer the question from a first-person perspective, rather than
saying what someone else said.
```

C.6.3 Evaluation Prompt

Single-Choice Question Template for the Evaluation Stage Prompt This template is utilised during the evaluation stage for answering single-choice questions. It incorporates {current_script}, the script related to the question extracted using RAG, and {dialog_history}, the dialog log relevant to the question, also extracted via RAG. {question} is the inquiry presented, and {choices} are the available options for that question.

```
Please answer the questions based on the information in your script and
the content of the dialog
Your Script relative to the question is {current_script}.
The dialog history relative to the question is {dialog_history}.
The question is {question}, the choices is {choices}
Let's think about this problem step by step, please provide your
reasoning and your choice (only the option number, e.g., 'a')
You must choose one from the options.
Please respond in the JSON format without any additional comments.
For example,
{
    "reason": "Your reason",
    "answer": "a"
}
```

Multiple-Choice Question Template for the Evaluation Stage Prompt Similar to the single-choice template, this format is designed for responding to multiple-choice questions during the evaluation stage.

```
Please answer the questions based on the information in your script and
the content of the dialog
Your Script relative to the question is {current_script}.
The dialog history relative to the question is {dialog_history}.
The question is {question}, the choices is {choices}
That is a multiple-choice question.
Let's think about this problem step by step, please provide your
reasoning and your choice (only the option number)
You must make a choice from the options.
Please respond in the JSON format without any additional comments.
For example,
{
  "reason": "Your reason",
  "answer": "a,b"
}
```

Single-Choice Question Template for FSA Prompt This template is specifically designed for Full Script Access (FSA), distinguishing it from conventional templates by granting access to the scripts of all players. Unlike standard procedures, it leverages the RAG technique to extract pertinent information from each player's script for comprehensive analysis.

```
Please answer the questions based on the information in each character's
script:
{current_script}
The question is {question}, the choices is {choices}
Let's think about this problem step by step, please provide your
reasoning and your choice (only the option number)
You must choose one from the options.
Please respond in the JSON format without any additional comments.
For example,
{
  "reason": "Your reason",
  "answer": "a,b"
}
```

Multiple-Choice Question Template for FSA Prompt Similar to the single-choice version, this template is tailored for answering multiple-choice questions under FSA conditions.

```
Please answer the questions based on the information in each character's
script:
{current_script}
The question is {question}, the choices is {choices}
That is a multiple-choice question.
Let's think about this problem step by step, please provide your
reasoning and your choice (only the option number)
You must make a choice from the options.
Please respond in the JSON format without any additional comments.
For example,
{
  "reason": "Your reason",
  "answer": "a,b"
}
```