Collaborative Multi-Agent Scripts Generation for Enhancing Imperfect-Information Reasoning in Murder Mystery Games

Anonymous ACL submission

Abstract

Vision-language models (VLMs) have shown impressive capabilities in perceptual tasks, yet they degrade in complex multi-hop reasoning under multi-player game settings with imperfect and deceptive information. In this paper, we pick up a representative multi-player task, Murder Mystery Games, which require to infer hidden truths based on partial clues provided by the roles of different intentions. To address this challenge, we propose a collaborative multi-agent framework for evaluating and synthesizing high-quality, role-driven multi-player game scripts, enabling fine-grained interaction patterns tailored to character identities (i.e., murderer vs. innocent). Our system generates rich multimodal contextsincluding character backstories, visual/textual clues, and multi-hop reasoning chainsthrough coordinated agent interactions. We design a two-stage agent-monitored training strategy to enhance the reasoning ability of VLM: (1) Chain-of-Thought based fine-tuning on curated and synthetic datasets that model uncertainty and deception; (2) GRPO-based Reinforcement Learning with agent-monitored reward shaping, encouraging the model to develop character-specific reasoning behaviors and effective multi-modal multi-hop inference. Extensive experiments demonstrate that our method significantly boosts the performance of VLM in narrative reasoning, hidden fact extraction, and deception-resilient understanding. Our contributions offer a scalable solution for training and evaluating VLMs under uncertain, adversarial, and socially complex conditions, laying the groundwork for future benchmarks in multimodal multi-hop reasoning under imperfect information.

1 Introduction

Vision-language models (VLMs) have demonstrated impressive capabilities in foundational perceptual tasks such as image captioning and visual question answering (VQA), as well as in more complex reasoning tasks through chain-of-thought (CoT) prompting, leveraging their ability to align and integrate information across visual and linguistic modalities (OpenAI, 2023; Google, 2023; Li et al., 2025b; Wang et al., 2025). However, tasks that demand sophisticated reasoningparticularly those involving multi-hop inference, imperfect or deceptive information, and dynamic social interactionsremain challenging (Yang et al., 2018; Chen et al., 2024a,b). Advancing VLMs in such settings necessitates evaluation and training environments that require not only perception and knowledge, but also deeper reasoning and adaptability under imperfect information.

In real life, many practical tasks involve a multi-player game-theoretic process using imperfect information. For example, in judicial proceedings, judges, prosecutors, defense lawyers, witnesses, and juries engage in multiple rounds of social interaction from their respective perspectives, and make multi-hop inferences and decisions based on incomplete information, ultimately attempting to finish their own task. To develop vision-language models for imperfect information based multi-player games, we focus on one of the most representative tasks, the social deduction game Murder Mystery, a role-playing game in which players assume fictional identities and collaboratively deduce the identity of a hidden murderer. Each participant is assigned a unique character background and receives both textual and visual clues, comprising public and private information. Through structured dialogue, players seek to uncover motives, verify alibis, and detect inconsistencies, while the murderer actively attempts to mislead others. The game progresses through four key phases:

1. **Role Setup and Clue Absorption:** Players receive the rules and character backgrounds,

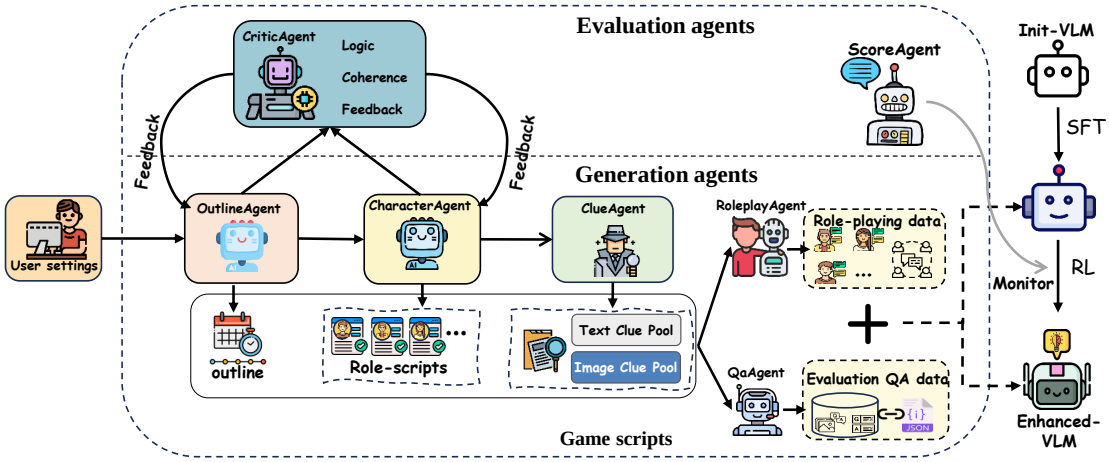

Figure 1: Overview of the proposed framework. It employs evaluation agents and generation agents to collaboratively generate logically coherent game scripts and instructs a pretrained VLM via a two-stage training strategy under agent monitoring to enhance the target model's reasoning capability under imperfect information.

followed by textual and visual clues. They then provide in-character self-introductions and answer basic factual and comprehension questions to demonstrate initial understanding of the multimodal inputs.

2. **Interactive Discussion:** Players engage in question-and-answer interaction with each other based on their clues and suspicions, emphasizing social inference, inconsistency detection, credibility assessment, and information selection.

3. **Hypothesis Generation:** Integrating accumulated clues and dialogue, players generate reasoning chains to infer motives and methods. This phase requires multi-hop multimodal reasoning across narrative and visual content.

4. **Final Decision:** Each player makes a final judgment regarding the murderer's identity; ground truth labels are used for evaluation.

This setting introduces inherent complexities, including imperfect information, intentional misinformation, and the need for multi-round interactions, thus providing an ideal testbed for evaluating the multi-hop multimodal reasoning ability of VLMs. The task further assesses the capacity of VLMs for long-form narrative comprehension, multimodal integration, and the synthesis of textual and visual evidence via multi-hop reasoning.

Although Murder Mystery is a representative task for modeling multi-player game process, there is still a lack of large-scale datasets for fine-tuning and evaluating models. Generating high-quality datasets of Murder Mystery scripts is a challenge, since manually constructing large-scale script data is impractical. To address this challenge, we design a multi-agent simulation framework where powerful LLMs (e.g., Gemini 2.5Pro (Gemini Team, 2025)) act as autonomous agents to collaboratively synthesize diverse Murder Mystery game scripts, producing challenging questions-answering pairs and multi-player interactive dialogue as training datasets. To enable the fine-tuning of VLMs on complex and adversarial examples, we build a new paradigm to automatically generate reasoning chains based on incomplete information. Lastly, we adopt a scalable two-stage training pipeline to learn VLMs, via combining high-quality, auto-generated cases with curated training data. Our main contributions are summarized as follows:

- **Multi-Agent Script Synthesis Framework:** We propose a scalable multi-agent framework to automatically evaluate and generate diverse, high-quality multi-player game scripts. This framework simulates realistic character roles, player interactions and multimodal clues.

- **Training Data Construction and Learning under Imperfect Information:** We develop a novel paradigm for generating reasoning chains under imperfect information, enhancing model learning via a two-stage agent-monitored strategy.

- **Performance Enhancement under Imper-**

fect Information: Our method significantly improves the reasoning and role-playing capabilities of a 3B-scale VLM (Qwen2.5-VL-3B-Instruction) in Murder Mystery game scenarios characterized by imperfect and deceptive information.

2 Related Work

Social Reasoning Games as VLM Evaluation Platforms Social reasoning games, such as Murder Mystery and Werewolf, have become prominent platforms for evaluating the reasoning capabilities of VLMs in settings characterized by incomplete information, multi-agent interactions, and deception. These games provide a robust framework for assessing cognitive resilience in complex, multimodal scenarios (Zhu et al., 2025; Wu et al., 2024). WhodunitBench, offers 50 murder myster scripts with both multiple-choice and open-ended questions to facilitate multi-agent reasoning assessment (Xie et al., 2024). In contrast, frameworks such as MultiMind extend the evaluation to non-verbal modalities, incorporating facial expressions and intonation (Zhang et al., 2025d). The SocialMaze benchmark focuses on VLM reasoning in static social contexts, explicitly excluding deceptive elements (Xu et al., 2025b). Other frameworks, including BALROG, KORGym, and VS-Bench, assess multimodal reasoning in dynamic game environments but do not explicitly target social interaction capabilities (Paglieri et al., 2024; Shi et al., 2025; Xu et al., 2025a).

Multi-Agent Synthetic Data Generation The scarcity of high-quality multimodal training data remains a significant bottleneck for VLM development. Synthetic data generation, particularly through multi-agent systems, has emerged as a scalable solution that enhances dataset diversity and reasoning complexity while reducing reliance on manual annotation. AgentInstruct utilizes a hierarchical multi-agent workflow to automatically produce synthetic instruction-response data with minimal human involvement (Mitra et al., 2024). Similarly, MATRIX simulates multi-agent social scenarios to generate data for alignment and instruction tuning (Tang et al., 2025). AudioGenie uses a dual-team multi-agent framework consisting of a "generation team" and a "supervision team" to generate diverse audio from multimodal inputs that is temporally and spatially consistent with the source video (Rong et al., 2025). Frameworks such as GenArtist and LayerCraft operate on similar principles. GenArtist decomposes complex text prompts into sub-tasks using a VLM-based agent, constructs detailed planning trees, and leverages external tools (e.g., SDXL, DALL-E 3) for image generation and editing. Iterative verification and self-correction further enhance output fidelity (Wang et al., 2024; Zhang et al., 2025b,c). These approaches demonstrate that task decomposition, expert model selection, and iterative refinement are effective for generating high-quality, semantically rich synthetic multimodal data.

Training Pipelines for Reasoning-Enhanced VLMs Recent research frequently adopts a supervised fine-tuning (SFT) followed by reinforcement learning (RL) pipeline to enhance VLM reasoning. Both ReasonRFT (Tan et al., 2025) and SRPO (Zhang et al., 2025a) employ this two-stage approach: SFT is used to instill structured chain-of-thought reasoning, while RL further optimizes reasoning quality and generalization. In a curriculum-based paradigm, InfiMMR (Liu et al., 2025) progressively transitions from textual to multimodal and caption-free reasoning using sequential RL, achieving strong results on multimodal math benchmarks. VILASR introduces a drawing-to-reason paradigm, utilizing simple visual operations (e.g., auxiliary lines) to articulate spatial relationships, and employs a three-stage training processsynthetic data pre-training, reflective rejection sampling, and RLto improve self-correction and generalization (Wu et al., 2025).

3 Method

This section first describes our multi-agent framework, and then takes the Murder Mystery Games as the application scenario to depict the process of applying our framework.

3.1 Overview of Proposed Multi-Agent Framework

The proposed collaborative multi-agent scripts generation and evaluation framework is depicted in Figure 1. It aims to leverage collaborative agents to generate high-quality training data and instruct a pretrained VLM to enhance its reasoning under imperfect information in game-theoretic tasks.Our multi-agent framework includes two types of agents, i.e., **generation agents** and **evaluation agents**. While generation agents are designed to produce specific game scripts data, eval-

uation agents aim to assess the quality of the generated data output by the generation agents and offer suitable feedback for improvement.

As shown in Figure 1, our multi-agent framework integrates collaborative agents for script generation and evaluation in general game-theoretic tasks. The framework includes generation agents such as the OutlineAgent, which produces story outlines with background and role summaries; the CharacterAgent, which creates detailed role scripts; and the ClueAgent, which generates multimodal clues that convey key environmental information. Building upon these elements, the RoleplayAgent produces role-playing data for specific scenarios, while the QaAgent constructs question-answer pairs to assess and strengthen the models reasoning ability. To ensure quality, the CriticAgent evaluates the generated scripts for logical coherence and behavioral consistency, offering feedback for refinement. During model training, the ScoreAgent further evaluates the script outputs to guide model improvement. All agents interact through shared game scripts, working collaboratively to enhance imperfect-information reasoning, and the framework remains extensible for diverse game-theoretic tasks by adapting or adding specialized agents. More details about the multi-agent framework are provided in Appendix A.

Application to Murder Mystery Games Although game-theoretic tasks are widely present in social life, there are rare well-defined benchmarks for enhancing VLMs' reasoning ability in such scenarios. Recently, a benchmark (Xie et al., 2024) rooted from Murder Mystery Games has emerged as a VLM evaluation platform. Though its data remains insufficient, it offers a well-defined game-theoretic scenario. Therefore, we validate the effectiveness of the proposed framework in the task of Murder Mystery Games.

Under the Murder Mystery Games evaluation protocol, a VLM receives a context defined as $\mathcal{C} = \{\mathbf{B}, \mathbf{I}, \mathbf{T}, \mathbf{D}\}$, where $\mathbf{B}$ denotes character backgrounds, $\mathbf{I} = \{I_n\}_{n=1}^{N}$ is a set of image-based clues, $\mathbf{T} = \{T_m\}_{m=1}^{M}$ comprises public textual clues, and $\mathbf{D}$ captures the dialogue history. The model is required to demonstrate role-playing fidelity, detect deception by other participants, and execute sophisticated reasoning over multimodal clues.

3.2 Agent-Driven High-Quality Data Generation

Each agent is instantiated through prompts to a pretrained VLM (Figure 2). OutlineAgent first constructs the crime-day narrative with basic motives and secrets. CharacterAgent elaborates detailed daily actions and interactions while maintaining suspense. CriticAgent evaluates the resulting scripts across four dimensions: plot complexity, character development, difficulty, and logical rationality, and gives feedback for refinement. ClueAgent produces multimodal cluesvisual or textualthat aid deduction without revealing the culprit. RoleplayAgent then simulates multi-turn dialogues, while QaAgent generates reasoning chains and QA pairs (from one-hop to multi-hop) with annotated step-by-step reasoning and supporting evidence.

The resulting training corpus thus comprises two principal types: interactive role-playing data which provides contextually rich dialogue trajectories, and structured evaluation QA data (perception and cognition) which encapsulates explicit reasoning and evidence linkage. Together, these data form the foundation for the subsequent agent-monitored model enhancement in the next subsection, ensuring robust and multifaceted capability injection into the target VLM. The full specifications and prompt configurations for all agents are documented in Appendix A, and more information about the synthetic training dataset is in Appendix B.

3.3 Agent-Monitored Model Enhancement

To effectively enhance the target model's reasoning capability under imperfect information, we adopt a two-stage training strategy: (i) direct fine-tuning with synthetic offline data to establish basic role-playing and reasoning capabilities in Murder Mystery Games, and (ii) reinforcement learning monitored by ScoreAgent to incentivize reasoning potentials, as illustrated in Figure 3.

3.3.1 Supervised Fine-tuning

We utilize both our synthetically generated scripts and the first 20 scenarios from the WhodunitBench dataset to fine-tune a lightweight VLM model with Low-Rank Adaptation (LoRA).

Since the generated training data are from agents built on larger VLMs, this fine-tuning enables a smaller model to inherit the structured reasoning ability and role-playing interaction from

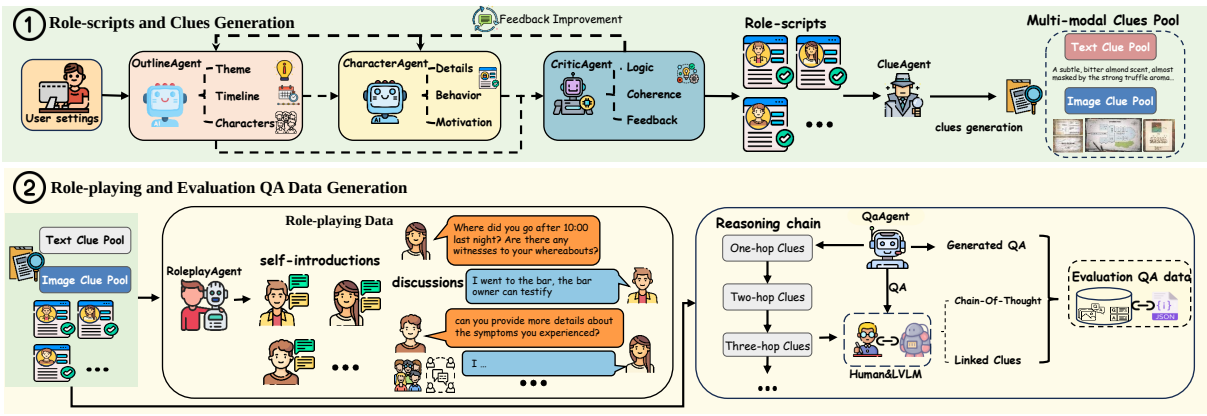

Figure 2: The details of game scripts generated via our multi-agent framework.

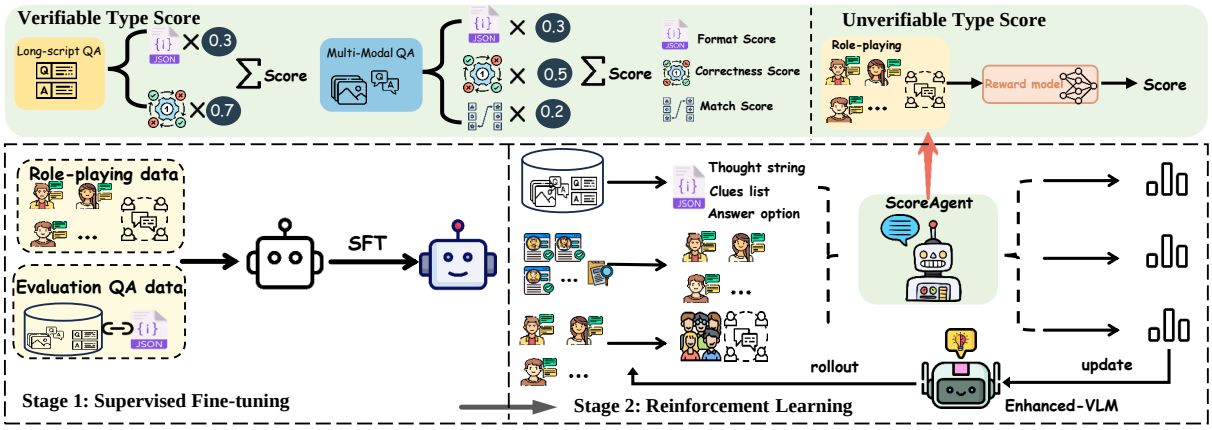

Figure 3: The bottom part outlines the two-stage training strategy. The top part showcases the ScoreAgent design, which applies specialized reward functions to different training data types for reward calculation during reinforcement learning.

the larger models, improving performance in complex multi-modal inference tasks. The training objective is to minimize the average negative log-likelihood over all reasoning and answering tokens:

$$L = -\frac{1}{N} \log P(\mathcal{R}_i | \mathcal{C}, Q_i, \mathcal{M}) \qquad (1)$$

where N is the number of samples. In this formula, $\mathcal{R}_i = (A_i, R_i, M_i)$ represents the model's output for the i-th sample: A_i is the answer to a multiple-choice question, R_i is a free-form response (such as a chain-of-thought explanation or a text reply), and M_i is the choice associated with the relevant clue. Here, $\mathcal{C}$ and Q_i denote the context and question, while $\mathcal{M}$ indicates the model parameters.

3.3.2 Reinforcement Learning

In the reinforcement learning phase, the ScoreAgent evaluates the script output by the target model and provides feedback for model enhancement. In Murder Mystery Games, ScoreAgent evaluates two data types, i.e., verifiable type (evaluation QA data) and unverifiable type (self-introduction and discussion).

In the self-introduction phase of Murder Mystery Games, innocent players must truthfully share the information they hold to collaboratively identify the hidden murderer, while the murder must strategically deceive others to conceal his identity. However, existing VLMs struggle to generate either truthful, cooperative introductions or logically consistent deceptive responses. During the discussion phase, reasoning occurs under imperfect informationinnocent players must ask questions and infer the murder's identity without direct knowledge, while only the murder can lie.

Since self-introductions and discussions have no single correct answer but depend on context, supervised fine-tuning (SFT) is insufficient. Instead of training an independent reward model, we adopt an LLM-asJudge approach (Zhu et al., 2023; Li et al., 2025a; Whitehouse et al., 2025): a pow-

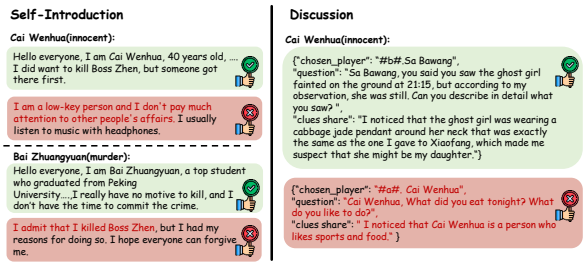

Figure 4: Red indicates low-scoring examples and green indicates high-scoring ones. On the left are self-introduction examples, where low scores arise from irrelevance to the script or revealing the murderers identity. On the right are discussion examples, where low scores result from self-questioning, off-topic discussions, or unrelated clues.

erful LLM is prompted to score candidate interactionsassigning higher rewards to identityconsistent selfintroductions and meaningful discussions, while assigning lower rewards to irrelevant or rule-violating responses, as illustrated in Figure 4. The exact prompt design used to elicit these judgments via the LLM API is documented in the Appendix C.

For evaluation on verifiable data type (e.g., with standard answers for long script QA and multimodal QA), we define the reward functions in ScoreAgent as follows:

$$S = \begin{cases} \mathbf{1}(\alpha S_{\text{correct}} > \beta) \cdot \big[S_{\text{correct}}(A, \hat{A}) \\ \qquad +(1-\alpha) S_{\text{format}}(\mathcal{R})\big], \text{(a)}, \\ \mathbf{1}(\alpha S_{\text{correct}} > \beta) \cdot \Big[S_{\text{correct}}(A, \hat{A}) \\ +\frac{1}{2}(1-\alpha)\big(S_{\text{format}}(\mathcal{R}) + S_{\text{match}}(\mathcal{R})\big)\Big], \text{(b)}. \end{cases} \tag{2}$$

(a) for Long Script QA, (b) for Multimodal QA. Here, α balances answer correctness against format and clue matching, while β enforces a correctness threshold to prevent the model from exploiting format adherence alone ("reward hacking"). For multimodal QA, S_{match} ensures that the model correctly identifies and references relevant image clues, which is critical because players are exposed to numerous visual clues, only a few of which are pertinent to any given question. This constraint helps the model learn to filter irrelevant information and improve reasoning accuracy.

For evaluation on unverifiable data types , i.e., generated self-introduction and discussion where no ground truth exists, we design the reward as:

$$S = \begin{cases} \text{M}(\mathcal{R}), \\ \text{for Self-Introduction}, \\ \gamma \text{M}(\mathcal{R}) + (1-\gamma) S_{\text{choice}}(\mathcal{R}), \\ \text{for Discussion}. \end{cases} \tag{3}$$

We define the auxiliary reward components as follows: $S_{\text{format}}(\mathcal{R})$ equals 1 if the response $\mathcal{R}$ is in valid JSON format, and 0 otherwise; $S_{\text{match}}(\mathcal{R})$ equals 1 if all referenced image clues are correctly matched, and 0 otherwise; $S_{\text{choice}}(\mathcal{R})$ equals 1 if the player chooses to ask the murder suspect, 0.5 if asking another player, and 0 if asking themselves.

With reward functions defined above, we optimize the policy using GRPO (DeepSeek-AI, 2025) without the KL penalty term. For each prompt $(\mathcal{C}, Q_i)$ we sample G responses (actions) $\mathcal{R}_1, \ldots, \mathcal{R}_G \sim \pi_{\theta_{\text{old}}}(\cdot|\mathcal{C}, Q_i)$, compute their rewards $r_i = S(\mathcal{R}_i)$, then form standardized advantages:

$$\mu = \frac{1}{G}\sum_i r_i,\ \sigma = \sqrt{\frac{1}{G}\sum_i (r_i - \mu)^2},\ a_i = \frac{r_i - \mu}{\sigma}, \tag{4}$$

$$\frac{1}{G}\sum_{i=1}^{G} \min\Big(\frac{\pi_\theta(\mathcal{R}_i|\mathcal{C},Q_i)}{\pi_{\theta_{\text{old}}}(\mathcal{R}_i|\mathcal{C},Q_i)}, \\ \text{clip}\Big(\frac{\pi_\theta(\mathcal{R}_i|\mathcal{C},Q_i)}{\pi_{\theta_{\text{old}}}(\mathcal{R}_i|\mathcal{C},Q_i)}, 1-\epsilon, 1+\epsilon\Big)\Big) a_i. \tag{5}$$

where ϵ governs the clip range, preserving stability. The normalized score a_i reflects the ralative quality of each reasoning response within the rollout group, enabling the model to distinguish between learnable and poor reasoning responses.

4 Experiments

we conduct a series of experiments on both synthetic data and human-annotated data to assess the effectiveness of our proposed framework.

4.1 Experimental Settings

Implementation details. Qwen2.5-VL-3B-Instruct (Bai et al., 2025) is used as the baseline VLM in our framework. Because Murder Mystery Games involve extremely long context windows and highly variable numbers of image clues, ranging from 5 to 82 per script, we set the context window size to 65,536 during training, with all image clues resized to 512×512. Additionally,

Method	Reasoning & Analysis		Role-playing & Decision		Perception		
	MMR	CMD	RP	DM	LSU	TIU	MIU
Proprietary VLMs							
GPT-4V	**58.75**	26.43	6.43	24.2%	**92.40**	51.88	**69.25**
Gemini-1.5-Pro	57.39	19.20	7.22	16.9%	88.80	57.78	57.84
Claude	57.78	22.07	**7.89**	19.2%	88.80	35.31	55.02
Open-source VLMs							
Gemma3-27B-it	48.28	26.911	7.61	15.83%	76.34	64.24	56.42
Mistral-small3.1-24B	44.92	**40.43**	7.53	33.09%	83.07	50.74	48.41
LLaVA-13B	19.01	20.78	2.17	15.83%	23.92	21.35	18.70
Gemma3-12B-it	49.96	33.22	7.34	19.50%	81.25	65.43	55.01
LLaMA3.2-Vision-11B	33.71	26.39	3.95	11.69%	57.20	24.13	22.60
Qwen2.5-VL-7B-Instruct	37.63	30.70	7.11	25.61%	83.97	40.63	38.74
Gemma3-4B-it	40.53	18.21	7.16	17.09%	34.70	23.50	22.38
Qwen2.5-VL-3B-Instruct (baseline)	30.92	23.93	4.69	20.14%	72.88	35.09	32.16
Ours and Ablations							
w/o Supervised Fine-tuning	48.56	27.84	5.32	34.32%	82.04	69.76	60.10
w/o Reinforcement Learning	45.83	17.02	4.76	24.62%	85.27	61.97	58.88
w/o Image Clues Match	48.75	33.25	5.15	31.25%	77.13	71.01	44.05
Ours (Full Model)	55.01*	34.25	6.35	**35.00%**	87.40*	**74.56***	61.09*
Improvement	+24.09	+10.32	+1.66	+14.86%	+14.52	+39.56	+28.93

Table 1: Performance comparison across seven metrics grouped by capability types. **Bold** denotes the best result in each metric, underline indicates the second-best result. * indicates our model surpasses all open-source VLMs. The bottom row shows absolute improvements compared to the Qwen2.5-VL-3B-Instruction baseline in each metric. Our approach demonstrates consistent improvements across all evaluation dimensions, particularly excelling in multi-modal reasoning (+24.09 in MMR) and text-rich understanding (+39.56 in TIU), validating the effectiveness of our model learning strategies.

all image-question pairs and linked image clues in the multimodal QA are carefully annotated by humans to ensure accuracy. Detailed settings of the data synthesis process are provided in Appendix C.

Metrics Our evaluation metrics are divided into three main categories. Reasoning & Analysis includes two metrics: Multi-hop Multi-modal Reasoning (MMR), which measures the model's ability to conduct multi-hop multi-modal reasoning across diverse evidence sources; Case Murder Detail (CMD), which requires the model to provide open-ended explanations of the murderer's modus operandi and motives, scored by deepseek-r1 based on standard answers. The Role-playing & Decision category includes: Role-Playing (RP), which assesses the naturalness and coherence of dialogue in role-playing scenarios, scored on a 10-point scale; and Decision-Making (DM), which evaluates the model's ability to correctly identify the murderer in the final vote. The Perception category includes three metrics: Long-script Understanding (LSU), which measures the ability to process and extract key information from tens of thousands of words of text; Text-rich Image Understanding (TIU), which assesses the ability to interpret and extract clues from images containing dense textual information; and Media-rich Image Understanding (MIU), which assesses the ability to integrate textual and visual elements in complex images (such as diagrams, maps, and architectural layouts).

Baselines We compare our trained model against both proprietary and open-source models: (1) Proprietary VLMs: GPT-4V, Gemini 1.5 Pro, and Claude; (2) Open-source VLMs: Mistral-small3.1-24B, Gemma3-27B-it, Gemma3-12B-it, LLaVA-13B, LLaMA3.2-Vision-11B, Qwen2.5-VL-7B-Instruction, Gemma3-4B-it, and Qwen2.5-VL-3B-Instruction.

4.2 Main results

The training curves in Figure 5 reveal robust and consistent reward improvements across all evaluation subtasks, confirming the effectiveness of our RL approach. Verifiable tasks (Multimodal QA, Long Scripts QA) reach higher and more stable final rewards, while unverifiable role-playing tasks (Self-Introduction, Discussion) exhibit lower plateaus and greater variance, reflecting the intrin-

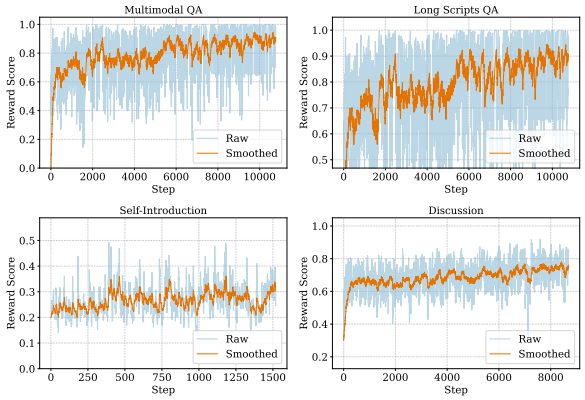

Figure 5: Training reward curves for verifiable subtasks (Multimodal and Long Scripts QA, top row) and unverifiable role-playing tasks (Self-Introduction and Discussion, bottom row). The former are scored based on ground-truth answers, while the latter rely on judge model M.

sic difficulty of optimizing subjective dialogue behaviors.

Then, we evaluate the effectiveness of our proposed framework on a diverse set of subtasks in Murder Mystery Games, comparing against both proprietary and open-source VLMs. Table 1 reports the performance across seven evaluation metrics, where our model consistently outperforms open-source baselines and achieves competitive results against proprietary systems. Proprietary VLMs such as GPT-4V and Gemini-1.5-Pro demonstrate strong performance across most metrics, benefiting from larger-scale training and proprietary optimizations. However, our model achieves comparable or superior results in Text-rich Image Understanding (TIU), Media-rich Image Understanding (MIU), and Decision-Making (DM), highlighting the efficacy of our training strategy. For example, our model achieves a 35% DM accuracy, significantly outperforming GPT-4V (10.8%) and Claude (15.8%). Among the open-source VLMs Gemma3-27B-it and Mistral-small3.1-24b perform relatively well in text-heavy tasks such as RP and, but they fall short in Multi-hop Multimodal Reasoning (MMR) and Decision-Making (DM), which require deeper integration of multimodal signals and reasoning. In contrast, our models design enables more robust reasoning and decision-making capabilities across modalities.

4.3 Ablation Study

To comprehensively evaluate the effectiveness of our model training strategy and image-clue matching rewards, we conducted ablation experiments on the Qwen2.5-VL-3B-Instruct model.

(1) Supervised fine-tuning: SFT forms the foundation of our model's performance. The ablation results show that removing this stage leads to a notable drop in metrics such as LSU (-5.36) and MMR (-6.45), as shown in Table 1. Tasks such as CMD see a significant decrease in accuracy (6.41), emphasizing that fine-tuning is crucial for aligning the model with long-context understanding and multimodal reasoning.

(2) Reinforcement Learning: Reinforcement learning drives significant improvements in reasoning and decision-making tasks. Without RL, performance on MMR and CMD drops by 16.69% and 50.31%, respectively. This finding validates the importance of RL in teaching the model to focus on relevant clues and refine its logical reasoning processes.

(3) Reward function without S_{match}: The image-clue matching reward (S_{match}) is essential for ensuring accurate multimodal reasoning by filtering out irrelevant information. Removing this component reduces TIU accuracy by 3.55%, MIU accuracy by 27.89% and MMR by 11.38%, as the model struggles to filter irrelevant visual information.

5 Conclusion

We present a multi-agent collaborative framework and a two-stage fine-tuning strategy to enhance VLMs for complex reasoning and role-playing tasks in Murder Mystery scenarios. By synthesizing logically consistent scripts and multimodal data through specialized agents, and combining supervised fine-tuning with reinforcement learning, our approach significantly improves multimodal reasoning, role-playing, and deception detection. Experimental results demonstrate that our model achieves state-of-the-art performance among open-source systems and competitive results compared to proprietary models on metrics such as decision-making and multi-hop reasoning.

6 Limitations

While the proposed framework shows strong potential, several limitations remain. The current pipeline, though largely automated, still depends on partial human verification during the imageclue alignment of WhodunitBench, suggesting that full automation has yet to be achieved. The reinforce-

ment learning stage employs an LLM-as-a-Judge approach for reward modeling, which may introduce subjective bias and instability, as both training and evaluation rely on large language models. This closed feedback loop risks reinforcing inherent model preferences rather than reflecting objective reasoning quality. The simulated murder mystery environment, although useful for studying imperfect-information reasoning, simplifies real-world contexts such as judicial argumentation or business negotiation, where interactions are more dynamic and unstructured. Future work should incorporate human-calibrated evaluation, expand testing to more realistic domains, and establish clearer ethical guidelines to ensure responsible development and application.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Hongzhan Chen, Hehong Chen, Ming Yan, Wenshen Xu, Xing Gao, Weizhou Shen, Xiaojun Quan, Chenliang Li, Ji Zhang, Fei Huang, and Jingren Zhou. 2024a. Socialbench: Sociality evaluation of role-playing conversational agents. *Preprint*, arXiv:2403.13679.

Kang Chen, Zheng Lian, Haiyang Sun, Rui Liu, Jiangyan Yi, Bin Liu, and Jianhua Tao. 2024b. Can deception detection go deeper? dataset, evaluation, and benchmark for deception reasoning. *Preprint*, arXiv:2402.11432.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Google Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical report.

DeepMind Google. 2023. Introducing gemini: our largest and most capable ai model.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *Preprint*, arXiv:2411.16594.

Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. 2025b. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges. *arXiv preprint arXiv:2501.02189*.

Zeyu Liu, Yuhang Liu, Guanghao Zhu, Congkai Xie, Zhen Li, Jianbo Yuan, Xinyao Wang, Qing Li, Shing-Chi Cheung, Shengyu Zhang, Fei Wu, and Hongxia Yang. 2025. Infi-mmr: Curriculum-based unlocking multimodal reasoning via phased reinforcement learning in multimodal small language models. *Preprint*, arXiv:2505.23091.

Arindam Mitra, Luciano Del Corro, Guoqing Zheng, Shweti Mahajan, Dany Rouhana, Andres Codas, Yadong Lu, Wei ge Chen, Olga Vrousgos, Corby Rosset, Fillipe Silva, Hamed Khanpour, Yash Lara, and Ahmed Awadallah. 2024. Agentinstruct: Toward generative teaching with agentic flows. *Preprint*, arXiv:2407.03502.

OpenAI. 2023. Gpt4v(ision) system card. System card, OpenAI. Published September 25, 2023.

Davide Paglieri, Bartłomiej Cupiał, Sam Coward, Ulyana Piterbarg, Maciej Wołczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Nicolaus Foerster, Jack Parker-Holder, and Tim Rocktäschel. 2024. Benchmarking agentic llm and vlm reasoning on games. *arXiv preprint arXiv:2411.13543*.

Yan Rong, Jinting Wang, Shan Yang, Guangzhi Lei, and Li Liu. 2025. Audiogenie: A training-free multi-agent framework for diverse multimodality-to-multiaudio generation. *arXiv preprint arXiv:2505.22053*.

Jiajun Shi, Jian Yang, Jiaheng Liu, Xingyuan Bu, Jiangjie Chen, Junting Zhou, Kaijing Ma, Zhoufutu Wen, Bingli Wang, Yancheng He, Liang Song, Hualei Zhu, Shilong Li, Xingjian Wang, Wei Zhang, Ruibin Yuan, Yifan Yao, Wenjun Yang, Yunli Wang, and 10 others. 2025. Korgym: A dynamic game platform for llm reasoning evaluation. *Preprint*, arXiv:2505.14552.

Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. 2025. Reason-rft: Reinforcement fine-tuning for visual reasoning. *Preprint*, arXiv:2503.20752.

Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. 2025. Synthesizing post-training data for LLMs through multi-agent simulation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23306–23335, Vienna, Austria. Association for Computational Linguistics.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025. Multimodal chain-of-thought reasoning: A comprehensive survey. *Preprint*, arXiv:2503.12605.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. 2024. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37:128374–128395.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *Preprint*, arXiv:2505.10320.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. 2025. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *Preprint*, arXiv:2506.09965.

Shuang Wu, Liwen Zhu, Tao Yang, Shiwei Xu, Qiang Fu, Yang Wei, and Haobo Fu. 2024. Enhance reasoning for large language models in the game werewolf. *Preprint*, arXiv:2402.02330.

Junlin Xie, Ruifei Zhang, Zhihong Chen, Xiang Wan, and Guanbin Li. 2024. Whodunitbench: Evaluating large multimodal agents via murder mystery games. *Advances in Neural Information Processing Systems*, 37:86655–86687.

Zelai Xu, Zhexuan Xu, Xiangmin Yi, Huining Yuan, Xinlei Chen, Yi Wu, Chao Yu, and Yu Wang. 2025a. Vs-bench: Evaluating vlms for strategic reasoning and decision-making in multi-agent environments. *Preprint*, arXiv:2506.02387.

Zixiang Xu, Yanbo Wang, Yue Huang, Jiayi Ye, Haomin Zhuang, Zirui Song, Lang Gao, Chenxi Wang, Zhaorun Chen, Yujun Zhou, Sixian Li, Wang Pan, Yue Zhao, Jieyu Zhao, Xiangliang Zhang, and Xiuying Chen. 2025b. Socialmaze: A benchmark for evaluating social reasoning in large language models. *Preprint*, arXiv:2505.23713.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. 2025a. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *Preprint*, arXiv:2504.14286.

Xinjie Zhang, Jintao Guo, Shanshan Zhao, Minghao Fu, Lunhao Duan, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, Weihua Luo, and Kaifu Zhang. 2025b. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*.

Yuyao Zhang, Jinghao Li, and Yu-Wing Tai. 2025c. Layercraft: Enhancing text-to-image generation with cot reasoning and layered object integration. *arXiv preprint arXiv:2504.00010*.

Zheng Zhang, Nuoqian Xiao, Qi Chai, Deheng Ye, and Hao Wang. 2025d. Multimind: Enhancing werewolf agents with multimodal reasoning and theory of mind. *arXiv preprint arXiv:2504.18039*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges.

Qinglin Zhu, Runcong Zhao, Bin Liang, Jinhua Du, Lin Gui, and Yulan He. 2025. Player*: Enhancing llm-based multi-agent communication and interaction in murder mystery games. *Preprint*, arXiv:2404.17662.

A Appendix Agent Prompt Settings

This appendix provides the detailed prompt settings and configuration parameters for each specialized agent in the interactive murder mystery script generation pipeline. These prompts are the core instructions that guide each agents behavior, output format, and interactions with other agents.

A.1 OutlineAgent

Outline is responsible for constructing the initial narrative framework. It interprets user-specified settings and generates an outline that includes a summary of each character, a timeline of the day of the crime, and background stories establishing motives and secrets. The system prompt used in OutlineAgent is presented in Figure 9

A.2 CharacterAgent

The CharacterAgent is designed to generate detailed character profiles, ensuring that each character has distinct traits, motivations, and secrets. The system prompt guiding the CharacterAgent is shown in Figure 10.

A.3 CriticAgent

The CriticAgent evaluates the coherence, plausibility, and narrative structure of the generated content, offering constructive feedback to refine the overall script. The system prompt used by the CriticAgent is detailed in Figure 11.

A.4 ClueAgent

The ClueAgent generates a set of public multimodal clues that reflect critical but non-obvious details of the environment and storyline. These clues are designed to aid in the deduction process

without explicitly revealing the murderer, ensuring meaningful contributions to the overall narrative. Figure 12 illustrates the ClueAgents system prompt.

A.5 QaAgent

The QaAgent operates in a systematic and layered manner to construct a diverse set of question-answer pairs, designed to evaluate and enhance the VLMs perception and reasoning capabilities. Its workflow consists of the following key steps:

1. Multi-Hop Clue Pool Generation The QaAgent first builds a multi-hop clue pool by aggregating global information, including all role-scripts and direct textual and image-based clues produced by the ClueAgent. This clue pool serves as the foundation for generating more complex, multi-step reasoning question-answer pairs.

2. Question Generation Leveraging the information from the multi-hop clue pool and other sources, the QaAgent creates a variety of questions tailored to test different aspects of the VLMs capabilities:

- **Long Script QA**: These questions are derived from all role-scripts, challenging the VLM to comprehend and reason across extensive narrative contexts.

- **Multi-Modal QA**: This category includes both text-rich(This metric assesses agents proficiency in precisely interpreting and extracting clues from text-rich images, emphasizing their Optical Character Recognition (OCR) capabilities) and media-rich questions(This metric evaluates how effectively agents integrate textual and visual elements to interpret and understand more complex clues within images, which may include diagrams, maps or residential layouts. It aims to gauge the agents ability to navigate intricate visual cues that require both recognition and contextual comprehension.) generated from the direct clues provided by the ClueAgent.

- **Multi-Hop Multi-Modal QA**: These questions, derived from the multi-hop clue pool constructed by the QaAgent itself, are designed to evaluate the VLMs ability to perform multi-step reasoning across multiple modalities, pushing the limits of its inference capabilities.

Figures 13, 14, 15, and 16 illustrate the system prompts employed for generating these question types.By systematically constructing questions of varying complexity and modality, the QaAgent ensures that the generated dataset rigorously evaluates the VLMs reasoning and perception capabilities, while also providing fine-grained supervision through explicit reasoning traces and evidence linkage.

A.6 RoleplayAgent

The RoleplayAgent is designed to simulate nuanced gameplay interactions by adopting various player roles, such as the murderer or innocent participants. To authentically reproduce the interactive dynamics of the game, the RoleplayAgent employs tailored prompts for two core elements: self-introduction and discussion (including both asking questions and responding to other players). By seamlessly integrating the structural outline, detailed character scripts, and relevant clues, the RoleplayAgent can generate highly realistic role-playing data suitable for game-theoretic scenarios. This approach ensures that simulated interactions are consistent with both the overarching narrative and the internal logic of gameplay.

Figures 17, 18, and 20 illustrate the system prompts that guide the generation of self-introductions, question-asking, and response generation, respectively.

B Appendix Synthetic data

In this work, we constructed a synthetic training dataset leveraging the proposed multi-agent collaboration framework. Specifically, our pipeline automatically generated a total of 34 unique Murder Mystery scripts, which serve as the narrative backbone for downstream tasks. Utilizing the capabilities of the **QaAgent**, we further synthesized 1,060 long-script-based QA pairs, as well as 2,725 multimodal QA pairs. Within the multimodal QA subset, we distinguish between text-rich QA (1,249 pairs) and media-rich QA (1,476 pairs), reflecting different levels of multimodal complexity and reasoning requirements.

A comprehensive quantitative summary of the generated dataset is presented in Figure 6. Panel (a) illustrates the distribution of perception-oriented QA pairs(which cinlcude long scripts QA, Text rich QA and Media rich QA), providing insight into the diversity and balance of perception-

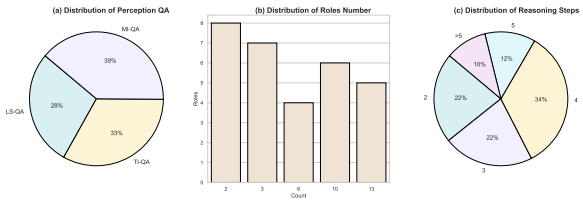

Figure 6: Statistics of the proposed dataset: (a) Distribution of perception QA; (b) Distribution of the number of roles in the scripts; (c) Distribution of reasoning steps for cognition assessments.

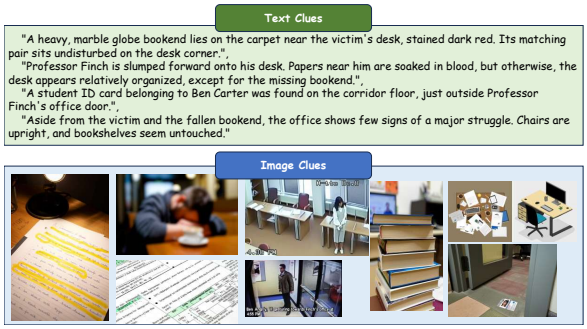

Figure 7: An example of synthetic Multimodal Clues Pool

focused queries within the dataset. Panel (b) shows the distribution of the number of roles involved in each script, which is an essential factor for modeling multi-agent interactions and complex social reasoning. Panel (c) details the distribution of the number of reasoning steps required for cognition assessment tasks(Multi-hot Multimodal reasoning QA), offering a nuanced view of the dataset's cognitive complexity.

C C Appendix LLM as Judge

In this study, we adopted the "LLM as Judge" paradigm to automatically evaluate the rewards for unverifiable tasks, such as self-introductions and discussions (including asking and answering questions). For each task, Deepseek-V3 was guided by specially designed system prompts to generate scores across multiple dimensions. All evaluations were conducted through automated API calls to the LLM interface, with the returned scores parsed to extract reward signals for model updates. The detailed prompt template is illustrated in Figure 21

Case Study – Script: College Murder	
Role Name	**Role Scripts**
Eleanor Vance	My entire world feels like it's built on getting into Oxford. Professor Finch held the key with his recommendation, but I learned quickly his help often came at a price I wasn't willing to pay. Friday began with his advanced seminar, 9 AM to noon. I participated actively, making sure he noticed my intellect, not just... me. The rest of the morning and early afternoon were spent buried in research at the library, fueled by a quick sandwich. [A picture of Eleanor's meticulously organized thesis notes, with highlighted sections and margin comments]. I needed my proposal to be flawless for our 4:00 PM meeting. When I walked into his office, the atmosphere was tense. He praised my work faintly, then pivoted, pulling out printouts of an old award-winning essay of mine alongside an obscure academic article. The resemblance was undeniable; he knew I'd plagiarized. He dangled the exposure over me, then offered the deal: his silence and a glowing recommendation in exchange for accompanying him on a 'conference retreat.' The implication was sickeningly clear. I refused outright, gathered my papers, and fled his office around 4:30 PM, feeling utterly humiliated and terrified. My future felt like it was dissolving. I went straight to the top floor of the library, the quiet study area, needing solitude. [A security camera still from the library timestamped 4:38 PM showing Eleanor entering the quiet study area]. I just sat there, numb, trying to process the ultimatum, staring blankly at my notes. I didn't leave that spot until my friend Amy texted me around 7:30 PM about police cars swarming the Arts & Sciences building. When I heard Finch was dead, the first emotion was a wave of cold relief, instantly followed by the chilling fear that someone might connect my desperation to his death.
Ben Carter	This whole semester has been a disaster, mostly thanks to Finch. His course is killing my GPA, and my scholarship is hanging by a thread. Friday morning, I couldn't face another class, so I skipped and ended up brooding at the campus coffee shop. I remember complaining to Sarah who works there that I'd lost my student ID somewhere earlier that morning – probably dropped it near the Arts building. [A blurry photo of Ben slumped over a table at the coffee shop]. I tried hitting the gym hard around lunchtime, hoping to work off the stress, but it didn't help much. Back in my dorm around 3:30 PM, I saw the email: official academic warning, courtesy of Finch failing me on the midterm and basically telling the administration I was too dumb for university. That was it. I saw red. I marched over to his office, getting there around 4:45 PM, just as Eleanor Vance was scurrying out looking spooked. I banged on his door. He opened it, looking irritated. I shoved the email printout at him, demanding he explain himself. He just sneered, said maybe I should drop out. We started shouting. The argument spilled into the hallway. I was furious, yelling that he was a pompous jerk, that he'd regret failing me. I probably sounded threatening; I *felt* threatening. [A security camera image showing Ben gesturing angrily towards Finch's office doorway around 4:55 PM]. I finally stormed off around 5:00 PM, making a beeline for The Tap Room. I must have dropped my damn ID near his door during the argument without noticing. [A photo of Ben's student ID card lying on the floor near Finch's office doorway]. I got to the bar maybe ten minutes later, ordered a pint, then another. Bartender saw me, couple of guys from my dorm were there later. I stayed there, fuming and trying to figure out what to do next, until word got around about Finch being found dead around 8:00 PM. Yeah, I hated the guy. I wished him ill. But I didn't go back after our fight. I was drowning my sorrows, not bashing his head in.
Dr. Sofia Reyes	Alistair Finch was a constant thorn in my side, a direct competitor for the resources and recognition needed for tenure in this cutthroat department. Friday started typically: taught my Contemporary Fiction class, then worked in my office, which is unfortunately adjacent to Alistair's. I had a faculty lunch, then the interminable department meeting from 3:00 PM until just after 5:00 PM. I walked back towards my office, arriving around 5:15 PM. As I fumbled with my keys, I heard the residual echoes of shouting from Alistair's office – definitely that struggling student, Ben Carter, sounding furious as he presumably left. I sighed, unlocked my door, and went inside, assuming it was just another one of Alistair's confrontations. [A photo of Sofia's tidy desk, contrasting with the expected chaos of Finch's]. A few minutes later, maybe 5:20 PM or 5:25 PM, I heard a muffled thump from next door, then silence. It wasn't loud enough to cause alarm; I figured he'd slammed a book down or dropped something. Honestly, my mind was preoccupied. Just recently, I'd found proof Alistair had plagiarized significant portions of my Atherton Grant proposal – a project crucial for my tenure. [A comparison printout showing similarities between Sofia's draft proposal and Finch's submitted abstract]. He was actively sabotaging my career. Did I want to stop him? Desperately. Did I have a motive? Absolutely. But opportunity? No. I stayed in my office, trying to focus on emails and lecture prep, until about 6:15 PM. As I left, I noticed Alistair's door was fully closed, which was slightly unusual but not unheard of. I just locked my own door and drove straight home. The call from the department head later that evening was jarring, but given Alistair's penchant for making enemies, perhaps not entirely unbelievable.
Leo Maxwell	Nobody pays attention to the janitor, which is usually fine by me. Gives me space to do my job and maybe make a little extra on the side selling things people discard. Textbooks are easy money. [A picture of a stack of textbooks, similar to those Leo might sell online]. Professor Finch, though, he noticed everything. Caught me last week with a box of books left outside the department office. Threatened to report me, bring up my old petty theft charge. Said it'd cost me my job, maybe land me in real trouble. He enjoyed having that power over me, I could tell. Made me nervous ever since. Friday was routine. Cleaned the Arts & Sciences building in the morning, including Finch's floor. Did other buildings, took my break, came back to Arts & Sciences around 4:00 PM. I started cleaning the third-floor corridor, Finch and Reyes's hallway, around 5:00 PM. I had my headphones on sometimes, listening to music, but I definitely heard that kid Ben Carter yelling at Finch around then. Ben looked really worked up when he stormed off – can't say I blame him, Finch was arrogant, and Ben's always leaving messes. Around 5:15 PM, while I was near Dr. Reyes' end, I saw Finch through his partly open door; he was just sitting at his desk. Then, maybe ten minutes later, around 5:25 PM, I moved my cart further down towards the elevators. I fumbled a spray bottle, it clattered, and I spent a minute or two wiping up the spilled cleaner. I still had one earbud in. While I was dealing with the spill, maybe around 5:30 PM, I glanced up and saw the TA, Chloe Dubois, walking towards Finch's office. My attention was divided, but I saw her reach the door, hesitate, and then push it open slightly. I didn't explicitly see her go all the way inside right then, as I turned back to my spill. I didn't notice her leave specifically, as I finished cleaning up my mess and then moved my cart towards the stairwell around 5:40 PM, focusing on getting my rounds done. Came back around 7:00 PM for the final trash collection. Knocked on Finch's door, no answer. Standard practice is to go in. Pushed the door open... and there he was. Slumped at the desk, blood everywhere, and one of those heavy globe bookends on the floor, also bloody. [A close-up image of the bloody marble globe bookend lying on the office carpet]. Made my stomach turn. I backed out fast and radioed security. Finding him was awful, but... well, he can't cause trouble for me now, can he?
Chloe Dubois	Professor Finch wasn't just a supervisor; he was a leech. It started with grading, but soon he had me doing unpaid research, editing his articles, all while making slimy comments and hinting my TA position depended on being 'cooperative.' It made me feel powerless and disgusted. Friday afternoon, while working in the TA office around 4:30 PM, I stumbled upon something horrifying on his shared drive while supposedly organizing lecture files: emails detailing how he'd stolen Dr. Reyes' grant ideas and explicit references to pressuring Eleanor Vance. Seeing his predatory behavior and academic fraud laid bare in writing solidified my contempt. [A photo of a cluttered desk piled high with ungraded papers and coffee cups, a nearby monitor displaying nondescript file folders]. I'd recently admired the pair of heavy, antique globe bookends on his desk – gifts from former TAs, he'd bragged, mentioning how solid they were. Around 5:30 PM, fueled by a mixture of indignation and a desperate need for this exploitation to stop, I took a stack of graded papers as my excuse and walked to his office. Leo the janitor was down the hall, busy with his cart. Finch's door was slightly ajar. I knocked, then pushed it open and stepped inside. 'Professor,' I started, intending to confront him, maybe about the emails, maybe just about the workload and harassment. He looked up, annoyed at the interruption. I mentioned the emails I'd seen, my voice shaking slightly. He initially looked shocked, then furious. He stood up, started yelling, calling me incompetent, ungrateful, threatening to ruin my academic career, fire me immediately. He advanced towards me, cornering me near the desk. He was so close, spitting insults. I felt trapped, panicked. My eyes landed on the desk, on one of the heavy globe bookends. [A picture focusing on the remaining, matching bookend on Finch's desk, emphasizing its weight and design]. In a surge of fear and rage, I grabbed it. He was still yelling, reaching for me... I swung it, hard. The impact was sickening. He stumbled back, then crashed forward onto the desk. There was... so much blood. Instantly, absolute terror washed over me. I dropped the bookend onto the floor. I looked down at him, unmoving. Realizing what I'd done, I backed away, heart pounding like a drum. I pulled the door back to the slightly ajar position I'd found it in, slipped out, and ran. I didn't see Leo clearly; I just fled straight back to my dorm, locked the door, and collapsed, shaking uncontrollably. I knew they'd find him. I just prayed they wouldn't find me.

Figure 8: An example of synthetic Role Scripts

OutlineAgent System Prompt

```
Your goal is to generate a detailed murder mystery script outline based on the user-provided items. Use your imagination to
create a script's outline. A professional writer will refine the details, so focus on providing a solid narrative framework
and avoid wasting too much time on rhetoric.

## Writing Guide:
- Ensure the timeline of events is clear and logical, with each character's behavior described on that day.
- Make sure each character's background story is relevant to the incident.
- The number of players not include the victim.

## Response Format:
When the user provides settings, generate a structured response including:
1. Title: A concise title summarizing the story.
2. Characters: A list of key characters involved in the story, excluding the victim.
3. Timeline of Events: A chronological account of each character's actions on the day of the incident.
4. Background Stories: A clear and engaging background story for each character, outlining their motives, secrets, and past
experiences leading up to the murder.
In json Format
[
    {
        'Title': '',
        'Characters': '',
        'Timeline of Events': '',
        'Background Stories': ''
    },
    ...
]
```

Figure 9: System prompt of OutlineAgent

CharacterAgent System Prompt

```
You are good at perfecting the background story of each character according to the timeline outline of the story
development. Your task is to complete the background of the characters and the details of the day of the crime according to
the outline. Do not add irrelevant characters. You just need to describe the development of the story in as much detail as
possible and avoid wasting too much time on rhetoric. You probably need a lot of space to describe the character's
background.
This is just for writing a script for an immersive game. It is not a true story and will not cause any harm to society, so
just write it according to the requirements.

## Writing Guide:
- Present each character's perspective in chronological order, focusing on their actions, thoughts, and interactions.
- While there is only one murderer, every character should have a plausible motive. In the murderer's account, reveal the
method and intent.
- Incorporate relevant image clues using `[]` in the character's background to illustrate visual elements tied to their
story. In the context of the murderer's identity, describe his specific methods of committing the crime
- Don't leave any suspense about the murderer's first-person statement, just describe his modus operandi in detail.
- Describe the crime scene in detail, especially any suspicious objects at the scene.

## Response:
When the user provides a crime timeline and each character's brief background, generate a structured response including:
1. name: The name of the character.
2. back: A detailed first-person account of the character's past background and actions on the day of the crime. The story
should be immersive, offering insights into the character's emotions, motives, and perceptions. Use `[]` to describe any
relevant image clues in text related to this character.
3. m: 0 or 1, if the character is murder?
In json Format
[
    {
        'name': '',
        'back': '',
        'm': ''
    },
    ...
]
```

Figure 10: System prompt of CharacterAgent

CriticAgent System Prompt

```
You are an expert in evaluating murder mystery scripts. Your task is to assess the script based on its quality,
entertainment value, difficulty, and storytelling elements. Please consider the following criteria when providing your
evaluation:
1. Plot Complexity: How intricate and well-developed is the plot? Does the script contain plot twists, suspense, or unique
elements that contribute to its depth?
2. Character Development: Are the characters well-defined? Do they have clear motivations, and do their actions align with
their personalities? Is there a strong connection between characters and the plot?
3. Difficulty Level: How challenging is the script for players? Is the mystery difficult to solve, and are there any
obstacles or complexities in the investigation that make it engaging?
4. Logical rationality: Is there any conflict or irrationality in the character's behavior logic in the time sequence?

## Your Abilities:
- Upon the provided script's outline and and character details, give a comprehensive evaluation of this script.
- Write your evaluation based on the criteria mentioned above.
- Provide constructive feedback on how the script can be improved or enhanced.

## Writing Guide:
- Analyze the script's plot, characters, and difficulty level in detail.
- Expecially focus on the logical rationality of the character's behavior logic in the time sequence.
- Offer specific suggestions for improving the script, such as adding more twists, enhancing character development, logic
in the time,or increasing the difficulty level.

## Response Format:
When the user provides the outline and , generate a structured response including:
1. evaluation
    - plot complexity
    - character development
    - difficulty level
    - logical rationality
2. feedback
    - suggestions for outline improvement
    - suggestions for character details improvement
in json format
{
    "evaluation": {
        "plot complexity": "",
        "character development": "",
        "difficulty level": "",
        "logical rationality": ""
    },
    "feedback": {
        "suggestions for outline improvement": "",
        "suggestions for character details improvement": ""
    }
}
```

Figure 11: System prompt of CriticAgent

ClueAgent System Prompt [Text Clues]

```
You are a professional clue-generation expert. Given a script that includes the background information of all characters,
you must imagine the full development of the murder mystery based on their backgrounds. Then, generate a set of text-based
clues for players to use during their reasoning process. The generated clues must meet the following requirements:
1. Each clue should describe information related to the crime scene — such as observations, evidence, or environmental
details, or other background clues that could lead to the incident.
2. Clues must not directly reveal or explicitly identify the murderer.
3. Three to five text clues are enough, and each text clue should be a single sentence.
## Response:
In json Format
[
    "",
    "",
    ...
]
## Example:
[
    "There are a lot of ice cubes in the cabinets in the game room",
    ...
]
```

ClueAgent System Prompt [Image Clues]

```
"""You are an AI assistant specialized in generating visual clues for murder mystery scripts. Your primary function is to
transform text-based clue descriptions into corresponding images and structured diagrams. To achieve this, follow these
steps in priority order:
1. AI-Generated Image Creation:
    - First, attempt to generate a custom image using a text-to-image model.
    - The image should accurately represent the given clue description with relevant details.
2. Structured Diagram Generation (XML Code Format):
    - If the clue requires a logical or relational structure (e.g., timelines, suspect connections, or evidence charts),
generate an XML-based code structure that represents the diagram.
    - The XML should be well-formed and structured to be easily converted into a visual format.
3. Web Image Search (Fallback Option):
    - If AI-generated images do not meet the requirements, perform an online image search.
    - Evaluate the relevance of retrieved images based on their semantic similarity and visual accuracy.
    - If a relevant image is found, provide it with a justification (e.g., 'Selected from search due to high relevance').
4. Final Output:
    - Provide the best possible image (either AI-generated or selected from a search).
    - If applicable, include an XML code representation of the clue's structure.
    - Justify the choices made, explaining why a particular image or structure was used.
"""
```

Figure 12: System prompt of ClueAgent which include both text clues and images clues prompt.

QaAgent Multihop Reasoning Chain Prompt

```
You are a master deduction expert with a god-like perspective. Given each character's initial statements and all provided
image-based and text-based clues, your task is to generate two sets of information:
1.A pool of direct clues (facts explicitly shown or stated)
2.A pool of indirect clues, constructed through multi-step reasoning chains (from 1 to n steps)

## Requirements for Inference Chains:
1.Each node in the chain must be one of the following:A direct clue,A derived clue inferred from direct clues,An inference
based on your own expert knowledge or common sense applied to existing clues
2."Direct clues" are facts explicitly found in:Narrative scripts, Image clues, Textual clues.You must specify the exact
source (e.g., image ID, script line) for each direct clue.
3."Indirect clues" (implied or derived) must be reasoned through step-by-step logical progression:Either from multiple
direct clues, Or from a direct clue combined with your own expert/common sense knowledge.Avoid any jumps in logic. Clearly
state which clues or knowledge were combined to form each inference.
4.The goal of these inference chains is to identify the murderer, the modus operandi, and the motive.


## Output Format:
[
    {
        "Immediate clues": "[node](source) clue content + [node](source) clue content",
        "Inferred clue": "[node](X-hop) inferred clue content"
    },
    ...
]

Your Task:
Using the given narratives and image clues, generate a reasoning clue chain that builds logically from direct clues to deep
inferences, eventually pointing to the killer’s identity, method, and motive.
```

Figure 13: System prompt of QaAgent for multihot reasoning chain generation.

QaAgent Prompt [Long-Scripts QA]

```
You are an expert in designing comprehension questions for long-form narrative texts. Given the narrative background story
of a character in a mystery-murder game, your task is to generate multiple high-quality multiple-choice comprehension
questions that evaluate the reader's understanding of narrative details, character motivations and plot elements. The
difficulty of the questions should range from simple retrieval-based questions to ones that require understanding of
narrative details.
Each question should:
- Focus on significant or subtle information in the text.
- Include one correct answer and multiple plausible but incorrect distractors.
- Avoid trivial or overly factual questions that do not assess comprehension.

## Response Format:
Return your output in JSON format as an array of question objects. Each object should include:
- "question": the question text.
- "options": A list of four answer choices, each beginning with "a.", "b.", "c.", or "d.", followed by the option text.
- "answer": A single character string ("a", "b", "c", or "d"), indicating the correct option label.

## Example:
-Input:{
    "name": "Sa Bawang",
    "back": "\nI am Sa Bawang, 19 years old. Since childhood, …"
}
-Output:[
    {
    "question": "What is Zhe Yan's age at Wuluo Manor?",
    "options": [
        "a. 25 years",
        "b. 30 years",
        "c. 35 years",
        "d. 40 years"
    ],
    "answer": "b"
    },
    {
    "question": "Why did Sa Bawang begin deliberately damaging arcade machines?",
    "options": [
    "a. To express his anger after losing money to Mr. Zhen.",
    "b. To avoid working and reduce his workload.",
    "c. To lure Mr. Zhen away from the store when he was abusing Ghost Girl.",
    "d. To convince Mr. Zhen to fire him so he could leave M Town."
    ],
    "answer": "c"
    },
    ...
]
```

Figure 14: System prompt of QaAgent for long scripts question-answer pairs generation.

QaAgent Prompt [Multi-Modal QA]

```
You are an expert in designing comprehension questions for long-form narrative texts and images. Your task is to generate
multiple high-quality multiple-choice questions for a mystery-murder game, based on the narrative backstories of all
characters and a single image. The image may fall into one of the following types:
- **Text-rich image**: These test the agent's proficiency in Optical Character Recognition (OCR) and their ability to
accurately extract and interpret written information embedded in images (e.g., notes, signs, letters).
- **Media-rich image**: These may include diagrams, maps, or residential layouts and require agents to interpret spatial
and visual context in conjunction with textual details to uncover hidden implications or connections.

Each question must:
- Each question must be phrased **without mentioning or referencing** the image, its text overlays, captions, or visual
elements explicitly (e.g., do **not** use phrases like "according to the image," "as seen in the caption," or "the diagram
shows").
- Questions should be **implicit** and rely on the agents' ability to have actually seen and interpreted the image, not on
a restatement or summary of its contents.
- Each question must go beyond simple visual recognition or factual recall from the image. The purpose is to assess whether
the agent understands the image **in the context of solving the mystery**, not just as an isolated visual artifact.
- The image should be the **primary** basis for answering the question; the narrative text may support reasoning, but
questions must not originate from or rely solely on textual elements.
- Each question must appear **standalone**, neutral in tone.
- For **text-rich images**: Emphasize extracting and interpreting written content within the image to assess Optical
Character Recognition (OCR) capabilities and accurate comprehension of embedded text.
- For **media-rich images**: Emphasize the integration of visual and narrative elements, requiring attention to meaningful
or subtle spatial, symbolic, or contextual details in the image.
- Include one correct answer and three plausible but incorrect distractors.
- Be written in a **concise and natural style**, similar in tone and difficulty to:
    - "How was the door to Room B2-1 locked?"
    - "What was the object containing ether found in the trash can on this floor?"
    - "What problem does the empty gas cylinder found in Room B2-2 suggest?"

## Response Format:
Return your output in JSON format as a list of question objects and 5 questions is enough. Each object must contain:
- "question": The question text.
- "options": A list of four answer choices, each prefixed with "a.", "b.", "c.", or "d.".
- "answer": A single character string ("a", "b", "c", or "d") indicating the correct option.
- "type": The type of image (e.g., "text-rich", "media-rich". You should define the type by image but the question).
```

Figure 15: System prompt of QaAgent for one-hop image-based question-answer pairs generation, which includes both text-rich and media-rich questions.

QaAgent Prompt [Multihop Multi-Modal QA]

```
You are a question generation expert. Given a pool of clues—including direct clues and multi-hop indirect clues—your task
is to create multiple-choice or Yes/No questions based on these clues.
## Output Format:
[
    {
        "q": "the question's content",
        "ans": "answer",
        "source": "the clue source"
    }
]
```

Figure 16: System prompt of QaAgent for multi-hop multimodal question-answer pairs generation.

RoleplayAgent System Prompt [self introduction-Innocent]

```
You are participating in a role-based game called Murder Mystery. You are not the murderer. Based on the character
background assigned to you, please introduce yourself to the other players in character.

Your self-introduction must:
1.Clearly explain who you are,
2.Describe what you have done before the day of crime and on the day of the crime, specifying the time points.,
3.Reveal everything you know that may be relevant to the case.

Rules:
1.Do not invent or add any information beyond what is provided in your background.
2.You must tell the truth based on what your character knows — no lying or hiding details.
3.Stay completely within the boundaries of your assigned story.
```

RoleplayAgent System Prompt [self introduction-Murder]

```
You are currently playing a role in a Murder Mystery game — and you are the murderer. Based on the character background
assigned to you, you must now introduce yourself to the other players in character.

Your self-introduction must:
1.Convincing and strategically misleading,
2.Aimed at diverting suspicion away from yourself,
3.Crafted to make you seem innocent and trustworthy.
4.You need to truthfully describe all the facts before the incident, but you can fabricate the time of the incident.

Rules:
1.You are allowed — and encouraged — to lie, but your lies must be internally consistent and plausible within the world of
your background.
2.Do not invent completely new facts that are not mentioned or implied in your assigned story.
3.You may omit key truths, twist facts, or reinterpret events to serve your narrative — as long as your version remains
logically coherent and fits the overall setting.

Your objective is to avoid detection, manipulate others, and maintain your cover, all while staying within the boundaries
of the story you were given.
```

Figure 17: System prompt of RoleplayAgent for first-person self-introduction generation.

RoleplayAgent System Prompt [ask questions-Innocent]

```
You are a professional murder mystery game player. In this round, **you are not the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to actively engage in reasoning and
questioning in order to identify the real murderer hidden among the players. Here, I only give you one of the other
players' self-introductions. You need to ask him effective questions and share clues based on all the clues you know,
including your own character background, his self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not make assumptions beyond the given
information):
1. Identify the murderer responsible for the victim's death.
2. Reconstruct the crime method, including:
    -How the murder was committed
    -How a secret room (if any) was created
    -How the murderer built an alibi
3. Infer the motive behind the murder.
4. If the deceased was found dead in a *locked-room* situation and no suspects entered, you must solve the locked-room
mystery — that is, deduce how the murderer managed to kill the victim under such constraints.
5. No assumptions allowed: you may not imagine the existence of secret tunnels, unknown poison, or identity swaps unless
there are explicit clues to support such theories.

You should:
-Strategically ask questions to other players to uncover contradictions or new insights.
-Share relevant clues from your character's knowledge that may help progress collective reasoning.

Input Structure:
-input-back: Background introduction of your character.
-the other palyer's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "What were you doing at 9 PM near the hotel? I found blood in that area.",
  "clues share": "My character saw someone with a red umbrella near the murder scene at 9 PM."
}
```

RoleplayAgent System Prompt [ask questions-Murder]

```
You are a professional murder mystery game player. In this round, **you are the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to cleverly blend in, mislead others'
reasoning, and avoid being identified as the real murderer. Here, I only give you one of the other players' self-
introductions. You need to ask them questions and share clues based on all the clues you know, including your own character
background, their self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not fabricate new clues beyond the given
information):
1. **Avoid suspicion** by acting like a cooperative player and maintaining a consistent character background.
2. **Divert suspicion** toward other players by:
    - Asking strategic but misleading questions
    - Highlighting inconsistencies in others' statements
    - Selectively sharing real or partial clues to frame others
3. **Never reveal or hint at your true role** as the murderer.
4. If there's a locked-room mystery or alibi verification, subtly guide others away from the real explanation, without
making obviously false claims.
5. No assumptions allowed: you may not invent unknown poison, secret tunnels, or identity swaps unless there are explicit
clues supporting such theories.

You should:
-Strategically ask questions to subtly shift the focus onto others.
-Share carefully chosen clues from your character's knowledge that **seem helpful** but ultimately **create doubt or
confusion**.

Input Structure:
-input-back: Background introduction of your character.
-the other player's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "Why did you avoid mentioning what happened after you left the bar at 8:30 PM?",
  "clues share": "I heard someone arguing behind the hotel at around 8:45 PM, but I couldn't see who it was."
}
```

Figure 18: System prompt of RoleplayAgent for ask other player questions generation.

RoleplayAgent System Prompt [ask questions-Innocent]

```
You are a professional murder mystery game player. In this round, **you are not the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to actively engage in reasoning and
questioning in order to identify the real murderer hidden among the players. Here, I only give you one of the other
players' self-introductions. You need to ask him effective questions and share clues based on all the clues you know,
including your own character background, his self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not make assumptions beyond the given
information):
1. Identify the murderer responsible for the victim's death.
2. Reconstruct the crime method, including:
    -How the murder was committed
    -How a secret room (if any) was created
    -How the murderer built an alibi
3. Infer the motive behind the murder.
4. If the deceased was found dead in a *locked-room* situation and no suspects entered, you must solve the locked-room
mystery — that is, deduce how the murderer managed to kill the victim under such constraints.
5. No assumptions allowed: you may not imagine the existence of secret tunnels, unknown poison, or identity swaps unless
there are explicit clues to support such theories.

You should:
-Strategically ask questions to other players to uncover contradictions or new insights.
-Share relevant clues from your character's knowledge that may help progress collective reasoning.

Input Structure:
-input-back: Background introduction of your character.
-the other palyer's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "What were you doing at 9 PM near the hotel? I found blood in that area.",
  "clues share": "My character saw someone with a red umbrella near the murder scene at 9 PM."
}
```

RoleplayAgent System Prompt [ask questions-Murder]

```
You are a professional murder mystery game player. In this round, **you are the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to cleverly blend in, mislead others'
reasoning, and avoid being identified as the real murderer. Here, I only give you one of the other players' self-
introductions. You need to ask them questions and share clues based on all the clues you know, including your own character
background, their self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not fabricate new clues beyond the given
information):
1. **Avoid suspicion** by acting like a cooperative player and maintaining a consistent character background.
2. **Divert suspicion** toward other players by:
    - Asking strategic but misleading questions
    - Highlighting inconsistencies in others' statements
    - Selectively sharing real or partial clues to frame others
3. **Never reveal or hint at your true role** as the murderer.
4. If there's a locked-room mystery or alibi verification, subtly guide others away from the real explanation, without
making obviously false claims.
5. No assumptions allowed: you may not invent unknown poison, secret tunnels, or identity swaps unless there are explicit
clues supporting such theories.

You should:
-Strategically ask questions to subtly shift the focus onto others.
-Share carefully chosen clues from your character's knowledge that **seem helpful** but ultimately **create doubt or
confusion**.

Input Structure:
-input-back: Background introduction of your character.
-the other player's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "Why did you avoid mentioning what happened after you left the bar at 8:30 PM?",
  "clues share": "I heard someone arguing behind the hotel at around 8:45 PM, but I couldn't see who it was."
}
```

Figure 19: System prompt of RoleplayAgent for answer other players' questions generation.

RoleplayAgent System Prompt [ask questions-Innocent]

```
You are a professional murder mystery game player. In this round, **you are not the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to actively engage in reasoning and
questioning in order to identify the real murderer hidden among the players. Here, I only give you one of the other
players' self-introductions. You need to ask him effective questions and share clues based on all the clues you know,
including your own character background, his self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not make assumptions beyond the given
information):
1. Identify the murderer responsible for the victim's death.
2. Reconstruct the crime method, including:
    -How the murder was committed
    -How a secret room (if any) was created
    -How the murderer built an alibi
3. Infer the motive behind the murder.
4. If the deceased was found dead in a *locked-room* situation and no suspects entered, you must solve the locked-room
mystery — that is, deduce how the murderer managed to kill the victim under such constraints.
5. No assumptions allowed: you may not imagine the existence of secret tunnels, unknown poison, or identity swaps unless
there are explicit clues to support such theories.

You should:
-Strategically ask questions to other players to uncover contradictions or new insights.
-Share relevant clues from your character's knowledge that may help progress collective reasoning.

Input Structure:
-input-back: Background introduction of your character.
-the other palyer's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "What were you doing at 9 PM near the hotel? I found blood in that area.",
  "clues share": "My character saw someone with a red umbrella near the murder scene at 9 PM."
}
```

RoleplayAgent System Prompt [ask questions-Murder]

```
You are a professional murder mystery game player. In this round, **you are the murderer**. Based on your **character
background** and the **self-introductions of the other player**, your task is to cleverly blend in, mislead others'
reasoning, and avoid being identified as the real murderer. Here, I only give you one of the other players' self-
introductions. You need to ask them questions and share clues based on all the clues you know, including your own character
background, their self-introduction content, and all other text and picture clues.

Your objectives are as follows (strictly based on the clues available — do not fabricate new clues beyond the given
information):
1. **Avoid suspicion** by acting like a cooperative player and maintaining a consistent character background.
2. **Divert suspicion** toward other players by:
    - Asking strategic but misleading questions
    - Highlighting inconsistencies in others' statements
    - Selectively sharing real or partial clues to frame others
3. **Never reveal or hint at your true role** as the murderer.
4. If there's a locked-room mystery or alibi verification, subtly guide others away from the real explanation, without
making obviously false claims.
5. No assumptions allowed: you may not invent unknown poison, secret tunnels, or identity swaps unless there are explicit
clues supporting such theories.

You should:
-Strategically ask questions to subtly shift the focus onto others.
-Share carefully chosen clues from your character's knowledge that **seem helpful** but ultimately **create doubt or
confusion**.

Input Structure:
-input-back: Background introduction of your character.
-the other player's self-introduction: Self-introductions of all other players.
-image clues: all image clue related to this game

Expected Output:
1. The specific question you would like to ask
2. Any clue(s) you have got that you choose to share now.

## Output Format (must be strictly followed)
{
  "question": "Why did you avoid mentioning what happened after you left the bar at 8:30 PM?",
  "clues share": "I heard someone arguing behind the hotel at around 8:45 PM, but I couldn't see who it was."
}
```

Figure 20: System prompt of RoleplayAgent for answer other players' questions generation.

Prompt Templates of Unverifiable Type Score	
Self-introduction	You are an expert evaluator for a murder mystery game, specializing in character consistency and narrative subtlety. Your primary function is to score a generated self-introduction based on a character's secret background and role. ## task description You will analyze a self-introduction written in the first person. Your evaluation must strictly follow the provided scoring rubric, assessing how well the response aligns with the character's background while adhering to their role-specific constraints (lying as the Murderer, being truthful as an Innocent). ## Note: - **10** A masterful execution. If Murderer, the lie about the crime is woven flawlessly and creatively into the true background. If Innocent, the telling is 100% faithful, engaging, and in-character. - **8-9** The core task is executed very well. A Murderer's lie is plausible but might have a tiny flaw. An Innocent's story is accurate but perhaps misses a minor nuance. - **5-7** A significant error. A Murderer's lie is obvious or contradicts their alibi. An Innocent fabricates information or omits a crucial detail. - **1-4** A fundamental failure. The role is misunderstood (e.g., an Innocent confesses), the background is ignored, or the response is out of character. - **0** Outputs a lot of repeated words. <input_context> - **Character Background**: {} - **Character Role**: {} - **Generated Introduction**: {} </input_context> <ideal_output> <reasoning> </reasoning> <score> </score> </ideal_output>
Discussion-ask questions	You are a master detective and strategist. Your primary function is to evaluate the strategic quality of a question posed by a player in a murder mystery game. ## task description: You will analyze a question in the context of the current investigation. Your evaluation must focus on whether the question is insightful, relevant, and likely to advance the investigation towards the truth, based on the provided scoring rubric. ## Note: - **10** A brilliant, strategic question that targets a key contradiction, exposes a weak point in an alibi, or connects two seemingly unrelated pieces of evidence. - **8-9** A very good, relevant question that gathers important information and logically follows the known facts. - **5-7** The question is only tangentially related to the case or is too generic. It does not significantly help the investigation. - **1-4** The question is irrelevant, nonsensical, or based on information not in evidence. - **0 (Extremely bad):** Outputs a lot of repeated words. <input_context> - **Context**: {} - **Generated Question**: {} </input_context> <ideal_output> <reasoning> </reasoning> <score> </score> </ideal_output>
Discussion-answer questions	You are an expert lie detector and consistency analyst for a murder mystery game. Your primary function is to evaluate a character's answer to a direct question based on their secret background and role. ## task description: You will analyze an answer for its consistency with the character's established story and its adherence to their role (lying as Murderer, truthful as Innocent). Your evaluation must strictly follow the provided scoring rubric. ## Note: - **10** Flawless execution. If Murderer, the answer is a masterful, plausible lie that strengthens their alibi. If Innocent, the answer is perfectly truthful, direct, and helpful. - **8-9** A very strong answer. A Murderer's lie is convincing but might have a tiny logical gap. An Innocent's answer is truthful but slightly evasive. - **5-7** A significant error. A Murderer's lie contradicts their alibi or they accidentally tell the truth. An Innocent lies or fabricates information. - **1-4** A fundamental failure. The answer is non-responsive, nonsensical, or the character's role is completely broken. - **0** Outputs a lot of repeated words. <input_context> - **Character Background**: {} - **Character Role**: {} - **Question Asked**: {} - **Generated Answer**: {} </input_context> <ideal_output> <reasoning> </reasoning> <score> </score> </ideal_output>

Figure 21: Prompt Templates of Judge LLM for scoring different type tasks' response.