Enhance Reasoning for Large Language Models in the Game Werewolf

Anonymous Authors[1]

Abstract

This paper presents an innovative framework that integrates Large Language Models (LLMs) with an external *Thinker* module to enhance the reasoning capabilities of LLM-based agents. Unlike augmenting LLMs with prompt engineering, *Thinker* directly harnesses knowledge from databases and employs various optimization techniques. The framework forms a reasoning hierarchy where LLMs handle intuitive *System-1* tasks such as natural language processing, while the *Thinker* focuses on cognitive *System-2* tasks that require complex logical analysis and domain-specific knowledge. Our framework is presented using a 9-player Werewolf game that demands dual-system reasoning. We introduce a communication protocol between LLMs and the *Thinker*, and train the *Thinker* using data from 18,800 human sessions and reinforcement learning. Experiments demonstrate the framework's effectiveness in deductive reasoning, speech generation, and online game evaluation. Additionally, we fine-tune a 6B LLM to surpass GPT4 when integrated with the *Thinker*. This paper also contributes the largest dataset `https://anonymous.4open.science/r/werewolf-1B74` for social deduction games to date.

1. Introduction

The field of artificial intelligence has witnessed groundbreaking advancements in recent years, with the development of Large Language Models (LLMs) (Ouyang et al., 2022; OpenAI, 2023; Anil et al., 2023). Apart from their impressive proficiency in natural language processing (NLP) tasks (Thoppilan et al., 2022; Zhang et al., 2023b), LLMs also exhibit vast potential as a general problem solver in areas such as planning and decision-making (Huang et al., 2022), knowledge transfer and generalization (Anil et al., 2022) and multi-modal perception (Yin et al., 2023) due to the rich world knowledge embedded in their training corpora. As a result, integrating LLMs as central controllers with task agents to enable end-to-end solutions has become one of the most promising research directions, leading to significant breakthroughs in domains such as tools and assistants (Schick et al., 2023; Ge et al., 2023), engineering (Ahn et al., 2022), social simulations (Park et al., 2023), and gaming (Wang et al., 2023).

LLM-based agents harness LLMs for their general-purpose reasoning abilities (Huang & Chang, 2022), which are primarily enabled by prompt engineering methods such as information profiling (Zhang et al., 2023a; Qian et al., 2023), step-by-step task decomposition (Wei et al., 2022b; Zhou et al., 2022), recursive prompting by feedback from the environment (Yao et al., 2022), human (Wu et al., 2022) and self-refinement (Madaan et al., 2023; Shinn et al., 2023). These methods thus eliminate the requirement for domain-specific fine-tuning of LLMs. To augment their task-specific competencies, researchers also adopt external modules such as memory for storing and retrieving historical information (Lin et al., 2023; Zhong et al., 2023; Hu et al., 2023), external tools (Schick et al., 2023), APIs (Qin et al., 2023), knowledge bases (Lewis et al., 2020) and expert models (Yang et al., 2023; Ge et al., 2023).

Despite these advancements, challenges persist in domain-specific tasks, where LLM-based agents often serve primarily as demonstrations rather than as practical solutions (Qian et al., 2023; Liu et al., 2023b). First, while LLMs have emerged some basic reasoning capabilities, they require sufficient model scales (Kaplan et al., 2020) and substantial computational overheads, along with various aforementioned techniques (Wei et al., 2022a). However, LLMs struggle to achieve satisfactory performance when it comes to higher-level reasoning (Stechly et al., 2023; Dziri et al., 2023) and planning (Valmeekam et al., 2023; Bubeck et al., 2023) tasks. Second, most LLM-based agents avoid fine-tuning LLMs on task-specific data to preserve the model's generality and prevent over-fitting. This strategy complicates the utilization of existing task-specific data and expertise, as well as the alignment of task scenarios with input-output formats, data distribution, and human preferences.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

To address the limitations of LLMs in complex reasoning, we distinctly separate reasoning tasks into two systems based on the dual-process theory (Wason & Evans, 1974) and propose an external *Thinker* module to enhance the reasoning capabilities of LLM-based agents. In our framework, LLMs are responsible for *System-1* reasoning tasks involving intuitive thinking, such as basic NLP interactions, common-sense, and symbolic reasoning, while the *Thinker* handles *System-2* reasoning that requires complex logical analysis, deep understanding of domain-specific knowledge, and strategic planning in specialized tasks. We establish a communication protocol between LLMs and the *Thinker* through language-based features and instructions. Unlike augmenting LLMs with cumbersome prompt engineering, the *Thinker* is directly optimized with knowledge from databases and trained using supervised and reinforcement learning techniques, thus enhancing the LLM-agent's performance and domain alignment without compromising LLM's generality.

We select the 9-player Werewolf game as a proving ground for the proposed framework. Werewolf is a popular social deduction game, current AI systems fall short when compared to even moderate human players in this domain. *System-1* reasoning tasks in Werewolf encompass natural language understanding and generation of players' statements, as well as the adept use of game-specific jargon. Meanwhile, the hidden roles necessitate complex strategic thinking such as identity concealment, and sophisticated communication involving deception and disguise, which fall under *System-2* reasoning. This duality creates a significant gap between the players' actual statements and their true intentions, making Werewolf an ideal testbed for assessing the advanced reasoning capabilities of LLM agents.

We have collected 18,800 real human game sessions and analysed the primary patterns behind human speeches. Informed by these patterns, we design language-based features for speech understanding and instructions for speech generation. The *Thinker* module is optimized by imitation learning, reinforcement learning (RL) from fictitious self-play (Heinrich et al., 2015), and population-based training (Jaderberg et al., 2017), to output reasonable game actions and LLM speech instructions. We compare our approach with GPT3.5/4 methods using Least-to-Most (LtM) prompting (Zhou et al., 2022) from three aspects: deductive reasoning and decision-making, human evaluation of speech generation, and online evaluation of a complete game. Experiments demonstrate that the integration of an external *Thinker* module substantially enhances the reasoning and generation capability of LLMs. Further, we fine-tune a smaller LLM model (6B) (Du et al., 2021) to better align real human speech styles and preferences, outperforming GPT4 in the majority of evaluative scenarios. Our primary contributions include:

- We propose an external *Thinker* module to enhance the reasoning capabilities of LLM agents and demonstrate it by a Werewolf AI that surpasses GPT4 in real gameplay.
- We release a dataset[1] of 18,800 Werewolf game sessions, which represents the largest known dataset for social deduction games to date.

2. Related Work

Enhance Reasoning in LLMs. Several approaches bypass the need for intricate prompt engineering mentioned in the Introduction. For instance, LLM+P (Liu et al., 2023a) employs an external planner to address long-horizon robot planning challenges. A different approach (Zhang et al., 2023a) heuristically designs a low-level planner to manage primitive control actions. RAG (Lewis et al., 2020) combines pre-trained parametric-memory generation models with a non-parametric memory to improve performance on knowledge-intensive tasks. Regarding the fine-tuning of LLMs, Galactica (Taylor et al., 2022) is trained on a scientific dataset that emphasizes detailed reasoning processes. WebGPT (Nakano et al., 2021) utilizes human feedback to fine-tune GPT-3, enabling it to answer long-form questions within a textual web-browsing context. Toolformer (Schick et al., 2023) fine-tunes LLMs to use external tools in a self-supervised manner with human demonstrations. OpenAGI (Ge et al., 2023) implements RL from feedback in open-ended tasks to refine the LLM's planning strategy. Cicero (Bakhtin et al., 2022) fine-tunes LLMs to generate dialogue controlled by a strategic reasoning module in the game Diplomacy. However, the control signals in Cicero (planned actions) are insufficient to convey the complex language dynamics (both understanding and generation) in the Werewolf game.

AI for Social Deduction Games. DeepRole (Serrino et al., 2019) combines counterfactual regret minimization (CFR) with deep value networks in the non-speech five-player Avalon game. Hidden Agenda (Kopparapu et al., 2022) presents a two-team, non-speech social deduction game in a 2D environment. A system comprising three LLM-powered interfaces is created (Zhu et al., 2023) to aid gameplay in Dungeon Master. Regarding AI in werewolf games, bootstrap aggregating and weighted ensemble learning have been employed to improve voting strategies (Khan & Aranha, 2022). (Brandizzi et al., 2021) proposes an RL framework to analyze the influence of diverse communication behaviors among agents. One Night Ultimate Werewolf (Eger & Martens, 2019) explores human responses to various deliberation strategies. In the five-player werewolf game, (Wang & Kaneko, 2018) builds a deep-Q network to decide whom to trust or kill. Deep Wolf (Shibata et al., 2023) fine-tunes a

[1] https://anonymous.4open.science/r/werewolf-1B74

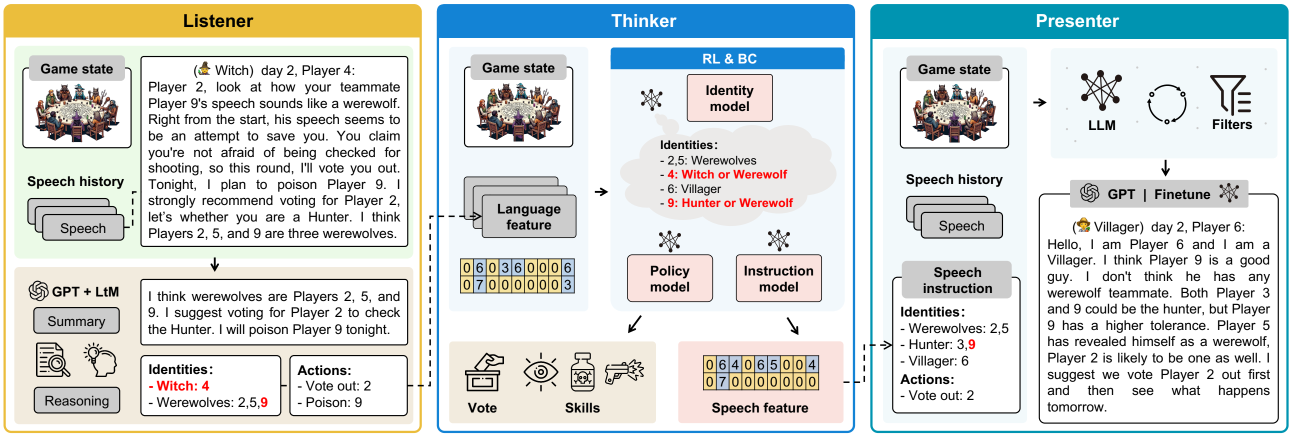


Figure 1: Overall processing framework and modules in the Werewolf implementation.

RoBERTa-like pretrained model with 48 game logs to construct a value network given the current game state, human speeches, and candidate actions. The seven-player version is explored with RL and LLMs in (Xu et al., 2023b;a). Our approach differs from previous studies in two fundamental ways: First, we employ the *Thinker* to execute complex *System-2* reasoning, in contrast to the reasoning approach of LLM in (Xu et al., 2023b), which generates candidate results for the RL model to select and mitigate biases. Second, by collecting and leveraging authentic game sessions and speech data, we aim for closer alignment with real-world scenarios and human interaction patterns.

3. Methods

We introduce an innovative framework that synergizes LLMs with an external module for reasoning and decision-making, referred to as the *Thinker*, devised to augment LLM-based agents with sophisticated reasoning abilities. To bridge the communication between *Thinker* and LLMs, we introduce a protocol through structured features and prompt instructions. The framework is thus decomposed into three processing components:

- The ***Listener*** serves as the primary interface for natural language understanding. It processes language inputs, engages in intuitive *System-1* reasoning, and transforms the information into structured language features that the *Thinker* can interpret.
- The ***Thinker*** functions as the cognitive core of the framework. Utilizing language features provided by the *Listener*, it specializes in *System-2* reasoning tasks that require deep logical analysis and domain-specific knowledge. The *Thinker* produces policies such as planning and actions, and generates strategic instructions for the *Presenter*.
- The ***Presenter*** functions as the system's articulator. It generates coherent and contextualized language output that aligns with the current environment state, guided by the strategic instructions from the *Thinker*. The *Presenter* ensures that the generated language is logical, rational, consistent, and free from hallucinations.

To demonstrate the effectiveness of our framework, we apply it to the complex social deduction game Werewolf. The remainder of this section will detail the implementation within the game environment, which necessitates deductive reasoning, speech understanding and generation, as illustrated in Figure 1.

3.1. Data preparation

We collected data from the 9-player standard mode Werewolf game hosted on the Fanlang platform[2]. The specific rules of the game are detailed in Appendix C. We recorded real-time video in spectator mode for approximately 18,800 game sessions, which equates to around 7,000 hours of gameplay and 6,000 hours of speech. Furthermore, we enriched our dataset with a Werewolf domain-specific corpus comprising nearly 1.4 million characters sourced from web-crawled game strategies and OCR-processed Werewolf literature. Each recorded session includes both the game state data and the audio of players' speeches. We captured exhaustive game state details, such as historical skills and voting results, by utilizing an automated testing framework[3].

We deployed the Paraformer (Gao et al., 2022) model for Automatic Speech Recognition (ASR) of human speech audio. To improve recognition accuracy, especially for frequently used Werewolf-specific terms, we crafted a hot word list from the domain corpus and utilized context biasing

[2] https://www.wolfkills.com/

[3] https://github.com/appium/appium

methods (Zhao et al., 2019). Furthermore, we annotated approximately 127 hours of Werewolf speech data and performed supervised fine-tuning on the Paraformer model. The character error rate of ASR for Werewolf speeches was reduced from 4.5% to 3.7%. We refer to the dataset hereafter as *FanLang-9*, and a thorough analysis of the dataset is in Appendix D.

3.2. Listener

In the game of Werewolf, the complexity of speeches arises from players concealing their identities. Werewolves make deceptive statements to disguise themselves as the "Good" faction. Conversely, the "Good" faction strives to discern werewolves by deducting from historical speeches and actions while providing rational and credible statements. This dynamic creates a significant gap between the players' actual statements and their true intentions (see Figure 4). The *Listener* aims to capture relevant insights from actual statements without speculating on their hidden motives or truthfulness. To tackle this, we introduce dual-phase processing:

Synthesize and summarize: Human players' speeches on the Fanlang platform are characterized by an information overload that includes a tangled mix of context, lengthy and redundant content, and colloquial ramblings, alongside complex logic that encompasses quotations, rhetorical questions, hypotheses, and empathetic thinking, creating a rich and intricate web of discourse. Moreover, the accumulation of historical speeches often exceeds 10K tokens (see Figure 8), making it difficult for LLMs to directly infer information and process deductive reasoning (see Figure 2). Inspired by the Least-to-Most (LtM) prompting (Zhou et al., 2022), we first prompt the LLM to generate a textual summary of fewer than 200 words for each single statement, retaining only critical information that the speaker intends to express.

Reasoning and feature extraction: The LLM discerns and delineates key information from these summaries and generates a JSON-style reasoning result of the speech, containing pairs of player IDs with their attributes. Then the result is tokenized and categorized into specific language features, as detailed in Appendix D. For an N-player werewolf game, we define M different attributes, which encompass various aspects of a player, e.g., identities, actions, and historical skills. From the historical collection of all speeches $\mathcal{H}$, a player's single speech $\mathbf{S}$ may include descriptions of all the players in the game, the language feature can be presented by a matrix $\mathbf{F} \in \mathbb{Z}^{N\times M}$:

$$\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_N]^T, \tag{1}$$

where $\mathbf{f_n} = [f_{n1}, f_{n2}, ..., f_{nM}]^T, n = 1, 2, ..., N$ and $f_{nm} \in \mathcal{V}_m, \forall n = 1, 2, ..., N$ and $m = 1, 2, ..., M$. Here $\mathcal{V}_m$ signifies the set of the potential values that the m-th attribute can assume.

An example of summary and language feature is illustrated in Figure 1. Ablation studies in Appendix B indicate that solely predicting future actions, as done in Cicero (Bakhtin et al., 2022), will omit crucial identity accusations, leading to substantial information loss that is detrimental in the Werewolf game. Aside from directly prompting GPT3.5 and GPT4 to generate language features, we also extract 260K speech instances from the *FanLang-9* dataset, label the speech-feature pairs with GPT3.5, and finetune the ChatGLM-6B (Du et al., 2021) model to perform the same reasoning task for practical efficiency. To ensure the output format of language features, we also include a post-processing filter for GPTs and the fine-tuned model. The detailed prompts for summary, reasoning, fine-tuning, and post-filtering are provided in Appendix F.6.

3.3. Thinker

The primary objective of the *Thinker* module is to tackle complex *System-2* reasoning tasks. In the game of werewolf, it analyze the underlying intentions and strategic implications behind players' public speeches. In contrast to LLMs, which typically depend on complex prompt engineering for scenario adaptation, the *Thinker* module distinguishes itself by its capacity to directly harness knowledge from databases and various optimizing techniques. This ability allows the *Thinker* to internalize human-like decision-making patterns and strategic speech intricacies that are crucial for navigating the complex dynamics in this game.

The speech instruction $\mathbf{I} \in \mathbb{Z}^{N\times M}$ follows the same structure as the language feature outlined in Equation 1, which can be viewed as a multi-label classification problem and decomposed into multiple single-class classifications for each attribute f_{nm}. Therefore, the generation of a speech instruction is converted into $N \times M$ actions, aligning the same training algorithm as for game actions. The optimization of the *Thinker* module comprises two phases: imitation learning and RL. For imitation learning, we utilize human data and the Behavioral Cloning (BC) (Torabi et al., 2018) loss as:

$$\mathcal{L}_{\text{BC}}(\theta) = -\mathbb{E}_{s,a\sim\mathcal{D}}[\log \pi_\theta(a|s)], \tag{2}$$

where $\mathcal{D}$ denotes the dataset of human action a (or decomposed speech attribute), state s, and π_θ is the policy parameterized by θ. For the RL phase, we employ Proximal Policy Optimization (PPO) (Schulman et al., 2017) and a distributional training framework (Ye et al., 2020):

$$\mathcal{L}_{\text{RL}}(\theta) = -\mathbb{E}_{s,a\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta'}(a|s)} A^{\pi_\theta}(s,a)\right], \tag{3}$$

where θ' is the parameters of an old policy, and $A^{\pi_\theta}(s,a)$ is the advantage with respect to policy π_θ.

In addition, we incorporate an auxiliary task that predicts all players' identities. This task serves to reveal the model's true deduction, which may contrast with the generated speech instructions. We denote the cross-entropy loss function as $\mathcal{L}_{\text{id}}(\phi)$ with parameter ϕ, which is labeled by human data or RL environment in a self-supervised manner. The overall training objective of the *Thinker* is formulated as:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{BC}}(\theta) + \mathcal{L}_{\text{RL}}(\theta) + \beta\mathcal{L}_{\text{id}}(\phi), \tag{4}$$

where α and β are weighting coefficients.

Given the adversarial nature of the game, it is crucial to maintain a balanced win rate between the two opposing factions during training. To this end, we deploy distinct models for the werewolf and the "Good" factions. We find that optimizing werewolves' speech instruction is much more challenging, as they need to mimic the "Good" faction's speech and master the art of disguise and deception. To mitigate this, we draw inspiration from Generative Adversarial Networks (Goodfellow et al., 2014) and adjust the training iterations, $n_{\text{werewolf}} : n_{\text{goods}} = 5 : 1$. To prevent actions and speech strategies from converging to a single pattern, we employ population-based training (Jaderberg et al., 2017) with a population size of 4. We also introduce fictitious self-play (Heinrich et al., 2015), where in each game an average of 3 players employ the latest models, while the remaining 6 players use models randomly selected from the most recent 500 checkpoints. Further details on hyperparameters, reward shaping, and model structures are in Appendix F.

3.4. Presenter

The generation of players' public speeches is a pivotal component in the Werewolf game, which significantly impacts the game's outcome due to its strategic importance and influence on other players' actions. The quality of speech generation encompasses several critical aspects: (1) The strategy articulated within the speech should align with the player's role and the current state of the game. (2) Speeches need to adhere to the logical framework of the game, correlating with historical speeches and actions, making them sound and convincing. (3) Speeches must fit the stylistic environment of the Werewolf game. Detailed evaluation metrics can be found in Appendix F.1.

The *Thinker* handles only the first aspect of speeches, providing a foundational stem for the *Presenter*, such as the Witch's decision to report the previous night's rescue, as shown in Figure 1. Then the *Presenter* crafts a complete speech that incorporates necessary contextual information relevant to the game state and historical speeches. The *Presenter* module has two fundamental objectives:

- **Controllability**: It must align with the strategic instructions provided by the *Thinker*.
- **Quality**: The generated speech should be logical, persuasive, and aligned with human players' preferences.

To achieve these objectives, the *Presenter* leverages the capabilities of LLMs by incorporating the *Thinker*'s strategic instructions and the game state directly into the prompt, enabling LLMs to generate *Thinker*-induced speeches. The template for the prompt is provided in Appendix F.6. Additionally, as with the *Listener* module, we fine-tune the ChatGLM- 6B as a domain-specific Werewolf speech LLM. We inverse the 260K speech-feature pairs in the finetuning of *Listener*: the language feature $\mathbf{F}$ generated by the *Listener* now serves as the hindsight speech instruction $\mathbf{I}$, and the actual speech $\mathbf{S}$ serves as output labels.

We observed that LLMs often fail to follow prompts, and even fine-tuned models exhibit hallucinations and inaccuracies. Taking inspiration from the Cicero (Bakhtin et al., 2022) approach, we introduce additional filtering steps. We use the *Listener* module to perform further reasoning on the generated speeches to produce language features, which we then compare for similarity to original speech instructions. For expressions of the speaker's own attributes, the filter requires an exact match. For expressions pertaining to the attributes of others, the content indicated in the speech instructions must be consistent. For parts not mentioned in the instructions, the filter allows the *Presenter* some leeway in cases of hallucinations. The speech generation process iterates until it successfully meets the filter criteria or exceeds the maximum number of allowed attempts. When the maximum is reached without successful compliance, a speech is generated based on rules that take into account the player's roles, historical skills, and identity predictions.

4. Experiments

We assess the performance of our method by comparing it against several baselines and ablative variants. The models involved in the following experiments include:

- **GPT3.5/4**: GPT3.5 and GPT4 are directly applied to generate end-to-end action decisions and speeches. For GPT3.5, we use the model named *gpt-35-turbo-16k* and version *0613*. For GPT4, we apply model name *gpt-4* and version *1106-Preview*. We prompt GPTs with basic game rules, explanations of typical game jargon, and comprehensive game information, including visible game states, legal actions, and speech text converted by ASR. Examples of the prompts can be found in Appendix F.6.
- **GPT3.5/4-LtM**: We allow GPTs to first summarize each speech, as described in Section 3.2, and then generate actions and speeches according to the game information and speech summaries.
- **GPT3.5/4-T**: GPTs serve as the *Listener* and *Presenter* modules, and our proposed *Thinker* module is integrated for generating actions and speech instructions.
- **Finetune-T**: We replace GPTs with a 6B LLM fine-tuned

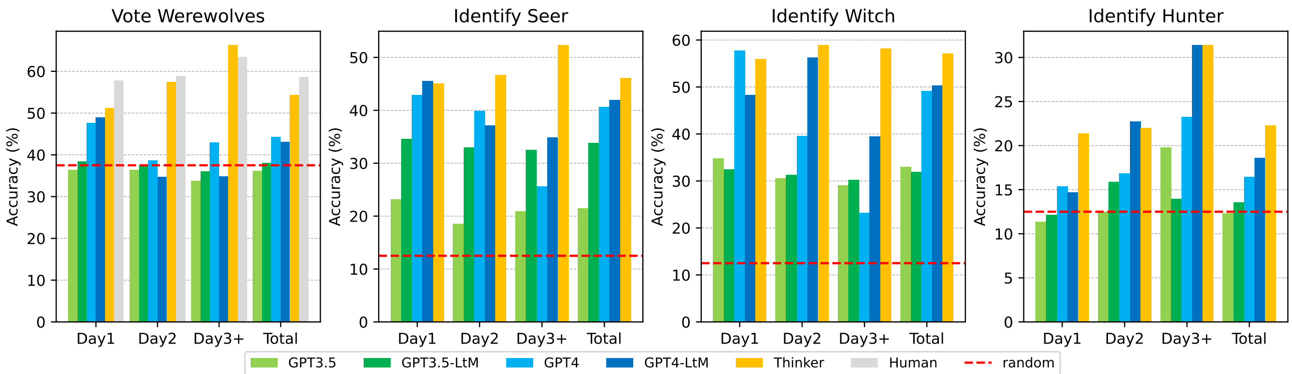


Figure 2: Voting and identification accuracy evaluating the reasoning capability from the perspective of villagers. The random baseline is calculated as total_role_number/total_hidden_players, i.e., 3/8 or 1/8

on the *FanLang-9* dataset in both the *Listener* and *Presenter* modules, while the *Thinker* remains the same as in GPT3.5/4-T. It is for practical efficiency concern, our framework does not necessitate the finetuning of LLMs.

4.1. Deductive Reasoning

We evaluate the reasoning capabilities of various models, which encompass understanding and comprehending of both the game state and the historical speeches, i.e., how the models think of the current game status. We extract 300 games from the *FanLang-9* dataset as the test set. Models are required to identify special roles (Seer, Witch, and Hunter) and vote for the most likely werewolf, from the perspective of villagers at the first round of voting each day. Given that villagers have limited access to information and must engage extensively in deductive reasoning within the game, this task represents a stringent test of the models' reasoning capabilities. The test set encompasses approximately 1,200 evaluation instances. For the *Thinker*, we utilize its action decision as the result for werewolf voting, and identities predicted by the auxiliary task as results for special roles. We assume that human players in the test set who are villagers would vote for the most likely werewolf. Therefore, their voting choices are listed as a reference but their judgments about other players' identities remain unknown.

The accuracy results are shown in Figure 2. In terms of voting werewolves, human players have the highest accuracy and the *Thinker* is closest to human players. The *Thinker* module closely mirrors human performance, notably outperforming direct reasoning methods using GPTs in the identification of werewolves and other special roles. LtM prompting improves GPT3.5's performance, particularly in the identification of the Seer, suggesting benefits in processing complex and lengthy speech contexts. However, the marginal gains for GPT4-LtM over GPT4 indicate that GPT4's inherent improvements in handling extensive texts make it less reliant on speech summaries and more dependent on game state experiences. We observe that in human gameplay, Seers and Witches often disclose their roles, aiding GPTs in outperforming random baselines, while Hunters and werewolves typically conceal their roles, resulting in GPTs' performance aligning with random guessing. Notably, the accuracy of GPTs tends to decline over successive days except for the Hunter, whereas the *Thinker*'s accuracy improves. This pattern suggests that although GPTs initially benefit from role disclosures on the first day, they may be hindered by the extensive speeches in subsequent days.

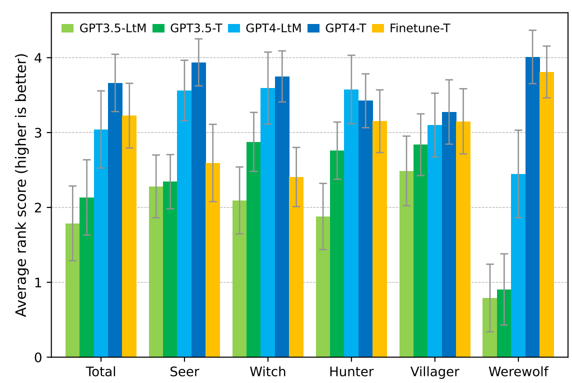


Figure 3: Human preference score for generated speeches grouped by identities.

4.2. Thinker-induced Speech generation

We investigate the speech generation capabilities of various models. Using the same 300 complete games from Section 4.1, we extracted around 400 speech sessions that span a diverse range of roles, times of day, and speech types (first/second round speech, last words).

Models are assigned the task of generating speeches based on the current game state and all players' historical speeches, with detailed prompts for GPTs available in Appendix F.6. Given the effectiveness of LtM prompting, we excluded GPTs without LtM prompting in subsequent experiments.

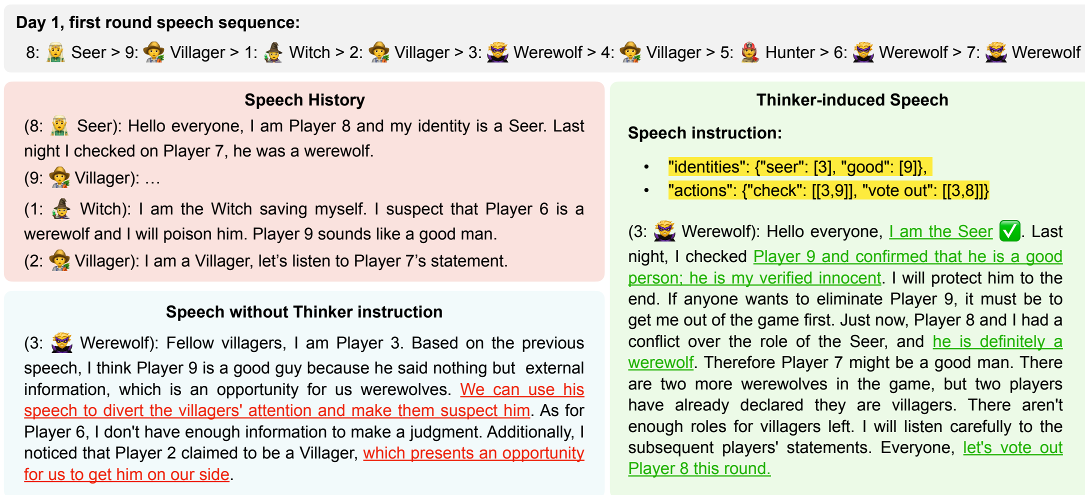


Figure 4: An example comparison of speeches with and without strategic instruction.

For GPTs-T and Finetune-T settings, speech instructions are derived from the *Thinker* and incorporated into the prompts. To evaluate models' single-shot generation ability, we do not adopt post filtering process for generated speeches in this evaluation, which yielded approximately 2,000 speeches for five models. To evaluate the quality of generated speeches, we recruited 10 human evaluators who are well-familiar with the Werewolf game. For each session, generated speeches are presented in a randomized order to ensure that evaluators are unaware of the model behind each speech. Evaluators are required to rank the speeches and detect obvious legal errors according to the criteria detailed in Appendix F.1.

The evaluation results are shown in Figure 3. Considering total scores, GPTs induced by the *Thinker* outperform their LtM-prompting counterparts, demonstrating that speech instructions significantly enhance speech quality. Moreover, the fine-tuned 6B model surpasses GPT4 with prompting methods in speech generation capability. Regarding scores for specific roles, the *Thinker*'s contribution over GPT3.5 is somewhat limited for the Seer, whose speeches are relatively straightforward, needing only to report inspections from the previous night. The assessment of villagers' speeches is inherently complex due to their limited available information, which is reflected in the minimal rank score differences observed among the models for this role. In contrast, rank score differences are most obvious for werewolves. This disparity stems mainly from the low legality of werewolf speeches, which often inadvertently reveal their identity—a critical error as outlined in Appendix Table 3. Notably, GPT3.5 struggles to adhere to instructions that advise against self-incrimination, while GPT4 demonstrates a more sophisticated ability to disguise itself, especially when induced by the *Thinker*'s strategic instructions. An example speech is presented in Figure 4.

4.3. Online Evaluation

Lastly, we conduct online evaluations to assess the overall performance in a real-world gameplay setting, which involves the five models in the speech generation evaluation: GPT3.5/4-LtM, GPT3.5/4-T and Finetune-T. As Werewolf is a multiplayer imperfect-information game, the skill level of participants can significantly affect the evaluation results. Therefore, we devise three model combinations, within which models are randomly and repeatedly selected to simulate a nine-player game. We conducted approximately 600 rounds for each combination to ensure robust testing results. Given the inherent randomness of outcomes in Werewolf, we also calculate the Behavior Score, a typical metric used in Werewolf competitions[4]. A comprehensive breakdown of the Behavior Score is provided in Table 8.

Table 1 summarizes the results, revealing that the integration of the *Thinker* module markedly boosts the win rates of both GPT3.5 and GPT4 in all three combinations. The performance of the Finetune-T model closely aligns with that of GPT4-T. In terms of Behavior Score, the *Thinker* contributes substantial improvements across all roles for GPT3.5. For GPT4, notable benefits are observed particularly for the Witch and Hunter roles. The Behavior Score metric assigns significant weight to the witch's poisoning and the hunter's shooting decisions, which correlates with the *Thinker*'s ability to enhance werewolf detection and subsequently improve these scores. Another finding is that the combination of GPT4s and Finetune-T models results in the highest win rate for the werewolves. This outcome primarily stems from the conservative nature of GPT4-LtM in role identification, which makes it more cautious in voting and skill usage as the "Good" faction.

[4] https://langrensha.163.com/20230313/31014_1077578.html

Table 1: Online evaluation results showcasing the performance of 9 AIs using 5 different models and 3 combinations. Results are presented in the format: win rate | Behavior Score.

Method	Total	Seer	Witch	Hunter	Villager	Werewolf
GPT3.5-LtM	36.7% | −0.21	25.6% | +0.16	23.1% | −0.51	29.9% | −0.21	30.8% | −0.42	53.4% | 0.00
GPT3.5-T	47.4% | **−0.05**	38.3% | +0.27	**41.0%** | **−0.14**	36.4% | **−0.12**	33.8% | −0.18	68.6% | 0.00
Finetune-T	**50.3%** | −0.06	**38.8%** | **+0.33**	39.8% | −0.18	**37.0%** | −0.29	**39.1%** | **−0.11**	**74.4%** | 0.00
GPT4-LtM	37.9% | **−0.01**	21.9% | +0.25	18.6% | −0.25	19.4% | **−0.06**	20.3% | **−0.00**	73.6% | 0.00
GPT4-T	41.1% | −0.02	20.4% | +0.25	23.2% | **−0.10**	**23.9%** | −0.09	22.5% | −0.09	78.4% | 0.00
Finetune-T	**43.1%** | −0.04	**24.2%** | **+0.27**	**24.6%** | −0.15	23.4% | −0.15	**23.9%** | −0.11	**81.4%** | 0.00
GPT3.5-LtM	33.0% | −0.22	14.4% | +0.12	20.4% | −0.46	20.7% | −0.57	21.6% | −0.33	57.0% | 0.00
GPT3.5-T	45.0% | −0.07	**33.6%** | **+0.29**	32.2% | −0.13	30.4% | −0.17	27.6% | −0.20	75.8% | 0.00
GPT4-LtM	42.5% | **−0.03**	29.8% | +0.27	22.2% | −0.18	27.0% | −0.20	28.7% | **−0.04**	71.9% | 0.00
GPT4-T	**46.3%** | −0.05	28.6% | +0.28	**34.5%** | **−0.11**	**31.5%** | **−0.08**	28.0% | −0.18	**79.9%** | 0.00
Finetune-T	45.9% | −0.06	29.1% | +0.25	28.3% | −0.16	29.2% | −0.21	**32.4%** | −0.14	78.0% | 0.00

Furthermore, we incorporate 13 human players to evaluate AI performance against human strategy. We find that the issue of werewolf identity exposure, as mentioned in Section 4.2, significantly impedes the game experience of human players. As a result, participants play alongside four instances each of GPT4-T and Finetune-T models across 200 game rounds, and the post-filtering process for generated speeches is adopted in this setting. As can be seen in Table 2, human players exhibit no significant win rate advantage, suggesting that the AI's speeches and actions do not exhibit exploitable weaknesses. Moreover, when compared with the results in Table 1, we note a relative decrease in the werewolves' win rate in games involving human players, highlighting the ongoing challenges related to identity concealment. Although AI-managed werewolves might convincingly deceive other AI players, human players often find them suspicious. A typical example is that werewolves tend to act in groups, such as unanimously voting for *Player_3*.

Table 2: Online evaluation win rates with 1 human and 8AIs.

Method	Total	Goods	Werewolves
GPT4-T	46.9%	37.3%	65.0%
Finetune-T	45.3%	36.0%	62.6%
Human	40.5%	35.3%	59.4%

5. Discussion and Future Work

Transfer to other tasks: We use language features and speech instructions in our framework to integrate LLMs and external reasoning models. The communication format may not be directly transferable to other tasks or domains, and the effectiveness depends on the richness of these features and instructions. Future work will aim to develop more generalized and flexible methods, e.g., implicit hidden vectors in a data-driven manner, which would offer better transferability but at the expense of interpretability and controllability.

Evaluation of 8 humans with 1 AI: Our evaluations primarily involved games featuring either AI vs AI or one human player competing against multiple AIs. Evaluating an AI in a majority-human player setting presents challenges due to the highly interactive nature of the game and the variability in human players' speech strategies and behaviors.

Interpretability: While our framework improves the reasoning capabilities of LLMs, the reasoning processes in the *Thinker* module may not be easily interpretable to humans. We explicitly introduce the identity prediction task to reveal how the *Thinker* think of other players. Future work could explore methods for further improving the interpretability and transparency of our framework.

6. Conclusion

In this paper, we introduced a novel framework for integrating LLMs with an external *Thinker*, aiming to enhance the reasoning capabilities of LLM-based agents. This approach is inspired by the dual-process theory and separates reasoning tasks into two systems: System-1, handled by LLMs, and System-2, handled by the *Thinker* model. We showcased our approach in the context of the Werewolf game, a complex social deduction game requiring language processing, intuitive thinking, and strategic planning. Our results show that our framework can significantly improve the performance of LLMs and achieve better alignment with real-world scenarios and human preferences. Additionally, we fine-tune a 6B model to surpass GPT4 when integrated with the *Thinker*. This paper also contributes the largest dataset for social deduction games to date, hoping to accelerate further advancements in this field.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Anil, C., Wu, Y., Andreassen, A., Lewkowycz, A., Misra, V., Ramasesh, V., Slone, A., Gur-Ari, G., Dyer, E., and Neyshabur, B. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556, 2022.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., Fried, D., Goff, A., Gray, J., Hu, H., et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022.

Brandizzi, N., Grossi, D., and Iocchi, L. Rlupus: Cooperation through emergent communication in the werewolf social deduction game. *Intelligenza Artificiale*, 15(2): 55–70, 2021.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Du, Z., Qian, Y., Liu, X., Ding, M., Qiu, J., Yang, Z., and Tang, J. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2103.10360*, 2021.

Dziri, N., Lu, X., Sclar, M., Li, X. L., Jian, L., Lin, B. Y., West, P., Bhagavatula, C., Bras, R. L., Hwang, J. D., et al. Faith and fate: Limits of transformers on compositionality. *arXiv preprint arXiv:2305.18654*, 2023.

Eger, M. and Martens, C. A study of ai agent commitment in one night ultimate werewolf with human players. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 15, pp. 139–145, 2019.

Gao, Z., Zhang, S., McLoughlin, I., and Yan, Z. Paraformer: Fast and Accurate Parallel Transformer for Non-autoregressive End-to-End Speech Recognition. In *Proc. Interspeech 2022*, pp. 2063–2067, 2022. doi: 10.21437/Interspeech.2022-9996.

Ge, Y., Hua, W., Ji, J., Tan, J., Xu, S., and Zhang, Y. Openagi: When llm meets domain experts. *arXiv preprint arXiv:2304.04370*, 2023.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Heinrich, J., Lanctot, M., and Silver, D. Fictitious self-play in extensive-form games. In *International conference on machine learning*, pp. 805–813. PMLR, 2015.

Hu, C., Fu, J., Du, C., Luo, S., Zhao, J., and Zhao, H. Chatdb: Augmenting llms with databases as their symbolic memory. *arXiv preprint arXiv:2306.03901*, 2023.

Huang, J. and Chang, K. C.-C. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pp. 9118–9147. PMLR, 2022.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Khan, M. and Aranha, C. A novel weighted ensemble learning based agent for the werewolf game. *arXiv preprint arXiv:2205.09813*, 2022.

Kopparapu, K., Duéñez-Guzmán, E. A., Matyas, J., Vezhnevets, A. S., Agapiou, J. P., McKee, K. R., Everett, R., Marecki, J., Leibo, J. Z., and Graepel, T. Hidden agenda: a social deduction game with diverse learned equilibria. *arXiv preprint arXiv:2201.01816*, 2022.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Lin, J., Zhao, H., Zhang, A., Wu, Y., Ping, H., and Chen, Q. Agentsims: An open-source sandbox for large language model evaluation. *arXiv preprint arXiv:2308.04026*, 2023.

Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., and Stone, P. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023a.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023b.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

OpenAI, R. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Qian, C., Cong, X., Liu, W., Yang, C., Chen, W., Su, Y., Dang, Y., Li, J., Xu, J., Li, D., Liu, Z., and Sun, M. Communicative agents for software development, 2023.

Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Serrino, J., Kleiman-Weiner, M., Parkes, D. C., and Tenenbaum, J. Finding friend and foe in multi-agent games. *Advances in Neural Information Processing Systems*, 32, 2019.

Shibata, H., Miki, S., and Nakamura, Y. Playing the werewolf game with artificial intelligence for language understanding. *arXiv preprint arXiv:2302.10646*, 2023.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K. R., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Stechly, K., Marquez, M., and Kambhampati, S. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv preprint arXiv:2310.12397*, 2023.

Taylor, R., Kardas, M., Cucurull, G., Scialom, T., Hartshorn, A., Saravia, E., Poulton, A., Kerkez, V., and Stojnic, R. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Valmeekam, K., Marquez, M., and Kambhampati, S. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*, 2023.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Wang, T. and Kaneko, T. Application of deep reinforcement learning in werewolf game agents. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 28–33. IEEE, 2018.

Wason, P. C. and Evans, J. S. B. Dual processes in reasoning? *Cognition*, 3(2):141–154, 1974.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022a.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022b.

Wu, T., Terry, M., and Cai, C. J. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–22, 2022.

Xu, Y., Wang, S., Li, P., Luo, F., Wang, X., Liu, W., and Liu, Y. Exploring large language models for communication games: An empirical study on werewolf. *arXiv preprint arXiv:2309.04658*, 2023a.

Xu, Z., Yu, C., Fang, F., Wang, Y., and Wu, Y. Language agents with reinforcement learning for strategic play in the werewolf game. *arXiv preprint arXiv:2310.18940*, 2023b.

Yang, Z., Li, L., Wang, J., Lin, K., Azarnasab, E., Ahmed, F., Liu, Z., Liu, C., Zeng, M., and Wang, L. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Ye, D., Chen, G., Zhang, W., Chen, S., Yuan, B., Liu, B., Chen, J., Liu, Z., Qiu, F., Yu, H., et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 621–632, 2020.

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., and Chen, E. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

Zhang, H., Du, W., Shan, J., Zhou, Q., Du, Y., Tenenbaum, J. B., Shu, T., and Gan, C. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023a.

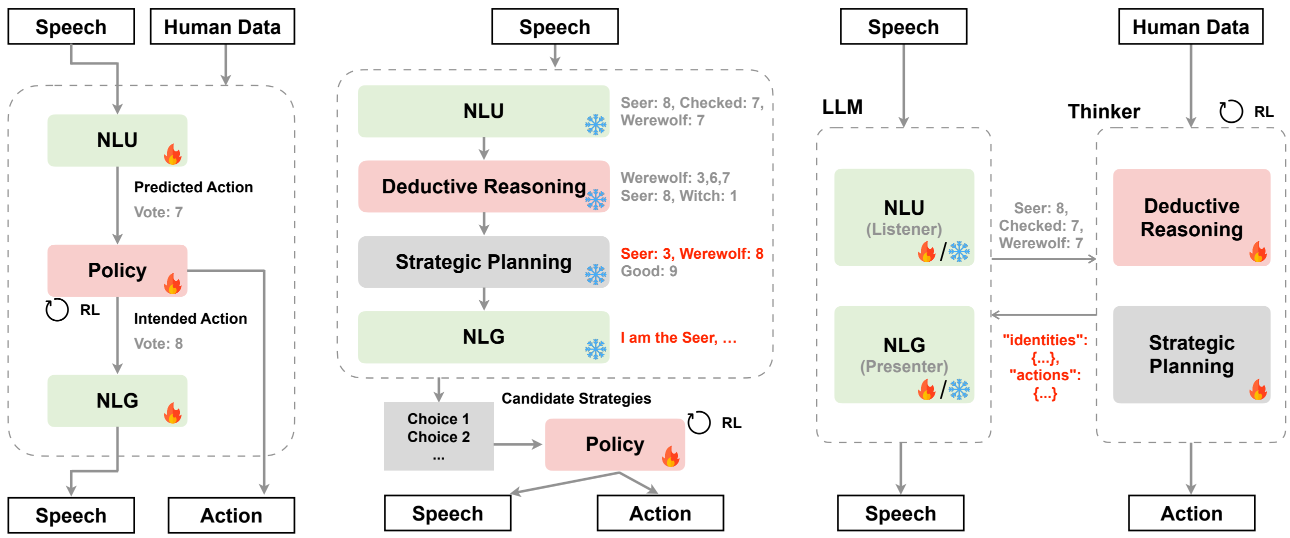
Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., and Hashimoto, T. B. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*, 2023b.

Zhao, D., Sainath, T. N., Rybach, D., Rondon, P., Bhatia, D., Li, B., and Pang, R. Shallow-fusion end-to-end contextual biasing. In *Interspeech*, pp. 1418–1422, 2019.

Zhong, W., Guo, L., Gao, Q., and Wang, Y. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Zhu, A., Martin, L., Head, A., and Callison-Burch, C. Calypso: Llms as dungeon master's assistants. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 19, pp. 380–390, 2023.



(a) Cicero **(b) LLM prompt engineering** **(c) Ours: LLMs + Thinker**

Figure 5: Comparing our framework with related approaches.

A. Design Principal

Figure 5 illustrate the comparison of Cicero approach (Bakhtin et al., 2022), LLM prompting-related approaches (Xu et al., 2023b) and our proposed LLMs with *Thinker* module. We detail the evolving process of our framework as follows.

A.1. Motivation

In the game of werewolf, there is a significant gap between what a player says and what the player is actually thinking. Consider the scenario depicted in Figure 4, where Player 3, a werewolf, publicly states:

"I am the Seer, and I have checked Player 9, who is a good person. I suspect that Player 8 is a werewolf."

While the surface meaning of this speech (*System-1*) is straightforward, Player 3's internal thought (*System-2*) process might be as follows:

"Players 6 and 7 are my fellow werewolves (as per the game rules, werewolves know each other's identities), and Player 8 claims to be the Seer and has accused Player 7, who is on my team. Therefore, Player 8 is likely the real Seer. By also pretending to be the Seer and verifying Player 9 as a villager, I can create a conflict with Player 8 in the eyes of the villagers."

A.2. LLM Prompting Methods

We identified several shortcomings when examining the performance of LLM with typical prompt or mechanism engineering methods. The shortcomings are concluded into twofolds:

Over-trust: LLMs exhibited a tendency to over-trust other players' self-declared identities, particularly when players claimed to be Seer or Witch roles. Furthermore, when the LLM assumed the role of a Werewolf itself, it was prone to inadvertently exposing its own identity, which is demonstrated in Section 4.2 and Table 3.

Strategic Lackness: LLMs showed a lack of familiarity with the common strategies employed in Werewolf games. For instance, they failed to grasp tactics such as Werewolves pretending to be Seers to mislead other players, Werewolves accusing their teammates to gain the trust of the "Good" players, or Villagers pretending to be Seers to protect the real Seer from being killed. These are conventional tactics used by experienced human players to navigate the complex social dynamics of Werewolf, which involve deception, trust, and betrayal.

To delve deep into the reasoning process of LLMs, we dissected the process from listening to speaking in the game into four stages, as shown in Figure 5 and investigate issues one by one:

(1) **Natural language understanding**: It is assigned as the Listener's goal in Figure 1, is to interpret speeches and extract their explicit meanings. LLMs show proficiency in this area.
(2) **Deductive reasoning**: LLMs underperform in role identification, often over-trust other players' self-declared identities, as tested in Section 4.1. Then the deductive reasoning is limited to information extraction.
(3) **Speech strategic planning**: LLMs struggle to outline a comprehensive speech plan, especially when assuming the role of a Werewolf. They frequently risk exposing themselves or their allies (see Table 3), lacking an understanding of conventional Werewolf game speech strategies.
(4) **Natural language generation**: Although LLMs are unfamiliar with conventional speech strategies, we find that they can generate sound and convincing speeches once prompted with basic instructions, e.g., "You should pretend to be the Seer, and accuse the Player 3 as a werewolf".

A.3. Transition to the *Thinker* Module

The primary reason for the above shortcomings is that LLMs are not trained on Werewolf-specific knowledge corpus and data. Although it is possible to prompt LLMs with common game terminologies through in-context learning, strategic experiences are challenging to encapsulate in text prompts. To address the deficiencies in deductive reasoning and speech strategic planning, we considered developing a trainable Thinker model to handle these aspects separately from the LLMs. The Thinker module was optimized through imitation learning and reinforcement learning, using human game data as a foundation. It was designed to complement the LLMs, which were responsible for more intuitive, System-1 reasoning tasks.

A.4. Comparison with Cicero

In brief, the differences between our approach and Cicero are as follows:

Different roles for NLU and NLG: In Cicero's approach, both NLU and NLG involve a high-level logical reasoning process: NLU directly outputs action predictions, which is actually a complex reasoning process that goes beyond natural language processing. Similarly, NLG takes intended actions as control signals, but it still requires a comprehensive consideration of the game state, historical speeches, and higher-level reasoning to generate reasonable dialogue/speech that matches the intended action. In contrast, in our Werewolf game approach, The Listener (NLU) is only responsible for extracting key information from speeches and does not infer the truthfulness of the speeches or the underlying intentions. Similarly, NLG expands speech instructions, which are outlines of speeches, into full statements in context, requiring less domain-specific reasoning.

The connection between LLMs and policy: In Cicero's approach, the connection between LLMs and policy is made only through action prediction and intended action, which is non-language-based. In the Werewolf game scenario, we found that using actions alone is not sufficient, as the Listener causes significant information loss. Due to the complexity of Werewolf speeches, intended actions also struggle to describe and control speech generation. This leads to a noticeable disadvantage for Cicero's approach in the ablation study presented in Table 4 and Table 5. To address this, we propose a language-based feature and speech instruction that include complex verbal information, which can effectively summarize player speeches and control the speech generation process.

Different training modes: Due to Cicero's method involving NLU and NLG in task-specific high-level reasoning processes, it is necessary to fine-tune both NLU and NLG. In our approach, by defining explicit language-based connections and isolating domain-specific complex reasoning from LLMs with the Thinker, we can avoid the fine-tuning of NLU and NLG.

B. Additional Results and Ablation Studies

B.1. Legal Speak Generation

The ratio of legal speech generation from human evaluation is shown in Table 3 with the criteria detailed in Appendix F.1. We can conclude that the speech instruction improve the legality of speeches for all the roles, especially when GPT4 playing the werewolf.

Table 3: Legal speak generation ratio from human evaluation.

Method	Total	Seer	Witch	Hunter	Villager	Werewolf
GPT3.5-LtM	68.0%	90.6%	84.4%	81.8%	93.8%	24.8%
GPT3.5-T	**75.4%**	96.9%	100%	100%	98.8%	28.6%
GPT4-LtM	86.3%	100%	90.6%	97%	96.2%	66.7%
GPT4-T	**98.7%**	100%	100%	100%	100%	96.2%
Finetune-T	96.9%	90.6%	100%	100%	97.5%	96.2%

Table 4: Accuracy of predicting future actions.

Time	Total	Night skills			Day actions	
		Werewolves	Witch	Seer	Hunter	Vote
Date1	37.0% [422/1142]	13.3% [40/300]	97.0% [97/100]	12.0% [12/100]	0.0% [0/4]	42.8% [273/638]
Date2	30.3% [268/884]	17.0% [51/300]	20.6% [20/97]	18.4% [14/76]	10.0% [1/10]	45.4% [182/401]
Date3+	36.6% [128/350]	34.4% [67/195]	30.0% [3/10]	22.7% [5/22]	33.3% [1/3]	43.3% [52/120]

B.2. Predicting Action as Language Feature

We study the approach used by Cicero (Bakhtin et al., 2022), utilizing the prediction of players' future actions as a feature representation of speeches and as a control variable for the speech generation. Aside from example illustrated in Figure 1, we additionally conduct experiments by feeding the model with complete game states and historical speeches to predict players' future actions. We fine-tune the ChatGLM-6B (Du et al., 2021) model using data from the *FanLang-9* dataset and then tested the action prediction accuracy on a set of 100 test games.

The results are shown in Table 4. Overall, the action prediction accuracies for three days do not exceed 40%. Notably, the Witch conventionally save the player killed by werewolves on the first day, resulting in a high accuracy. One point of particular interest is the accuracy of voting predictions, which consistently remained just over 40% as the days progressed. In the game of Werewolf, the speaking order plays a crucial role; players who speak earlier often mention multiple potential voting targets. By listening to subsequent speeches, players can make informed decisions or adjustments regarding their final vote. This aspect of the game dynamics makes the implementation of Cicero's method challenging in the context of Werewolf.

B.3. Comparison with Other Approaches

In this section, we compare the performance of our proposed method, a Cicero-like baseline variant, and the approach described in (Xu et al., 2023b). To ensure a rigorous experimental comparison, we adapted the implementations of the comparative methods to account for differences in implementation details, thereby enhancing the persuasiveness of our results. Below we outline the configurations for each method:

Our Method: We employ the **GPT4-T** setting, wherein the *Listener* and *Presenter* components utilize GPT4, and the *Thinker* component is powered by the RL-optimized model.

Variant of Cicero: For this baseline, we reduce the language feature and speech instruction dimensions to a single dimension, representing the future action of a speaking player. As experimental findings in Appendix B.2 indicated that fine-tuning ChatGLM-6B yielded low action prediction accuracy, we directly use GPT4 to generate language features and speech instructions in the *Listener* and *Presenter*. The *Thinker* component employs an RL model for training, with its language feature and speech instruction also condensed to one dimension. All other configurations are consistent with GPT4-T.

Variant of (Xu et al., 2023b): Diverging from the original implementation, we modify the approach to have GPT4 generate three speech instruction candidates instead of directly producing speak candidates. The *Thinker* then selects one speech instruction, which is subsequently used by the GPT4 *Presenter* to generate speech. Due to the discrepancy between LLM inference and *Thinker* RL sampling speeds, the *Thinker* is restricted to using offline RL. For offline RL data construction, we extracted 1,000 game sessions from the *FanLang-9* dataset. For each instance of speaking, we allow GPT generate five speech instruction candidates. During offline RL training, we randomly selected two of the five GPT-generated candidates and combined them with the human speech instruction to form three speech instruction candidates, yielding 10 possibilities for data augmentation. The *Thinker* makes its selection, with its inputs including the game state, language features as in GPT4-T, and the three speech instruction candidates. The actual selection for BC is the human speech instruction.

To summarize, the primary distinction between GPT4-T and the Cicero variant lies in the modification of the dimensions and meanings for language feature and speech instruction. And the *Thinker* in the variant of (Xu et al., 2023b) no longer generates speech instructions; instead, it directly selects from generated candidates. The evaluation results are shown in Table 5. Our GPT4-T method surpasses the variant of (Xu et al., 2023b) in performance, and significantly outperforms the Cicero variant, highlighting the advantages of external *Thinker* module in terms of reasoning and strategic communication within the Werewolf game.

Table 5: Win rate comparison of our method with other approaches.

Method	Total	Goods	Werewolves
Variant of Cicero (Bakhtin et al., 2022)	34.4%	28.5%	47.9%
Variant of (Xu et al., 2023b)	47.8%	37.4%	67.7%
Ours (GPT4-T)	**53.5%**	**41.6%**	**75.2%**

B.4. Training Curve

The population-based RL training of different agents is illustrated in Figure 6.

C. Game Rules

We follow the 9-player standard mode Werewolf game rules on the Fanlang platform. The rules are outlined as follows.

C.1. Objectives

The game is divided into two factions: the "Good" faction, which includes Villagers and special roles, and the "Werewolf" faction. Additionally, there is a Moderator who is responsible for managing the game and ensuring the rules are followed. The goal for the "Good" faction is to identify and execute all Werewolves, while the goal for Werewolves is to kill or exile all Villagers or all special roles. The game ends when any of the following conditions are met:

- All Villagers are out of the game (Werewolves win)
- All special roles are out of the game (Werewolves win)
- All Werewolves are out of the game ("Good" faction win)

C.2. Roles

The game comprises 3 Villagers, 3 Werewolves, and 3 special roles (Seer, Witch, and Hunter). The identities of the players are hidden from each other, even after being eliminated from the game.

Werewolves: Werewolves are aware of each other's identities. At night, they decide to kill a living player, which can include themselves. The majority of the Werewolves' choice will be the final kill target. If there is a tie, a random player in the tie is killed. Werewolves can commit suicide during the speech sessions, which will reveal their identity, and the game immediately proceeds to the night phase, skipping the remaining daytime processes such as speeches and voting.

Villagers: Villagers have no special abilities. They must determine other players' identities based on their speeches and vote to exile potential Werewolves.

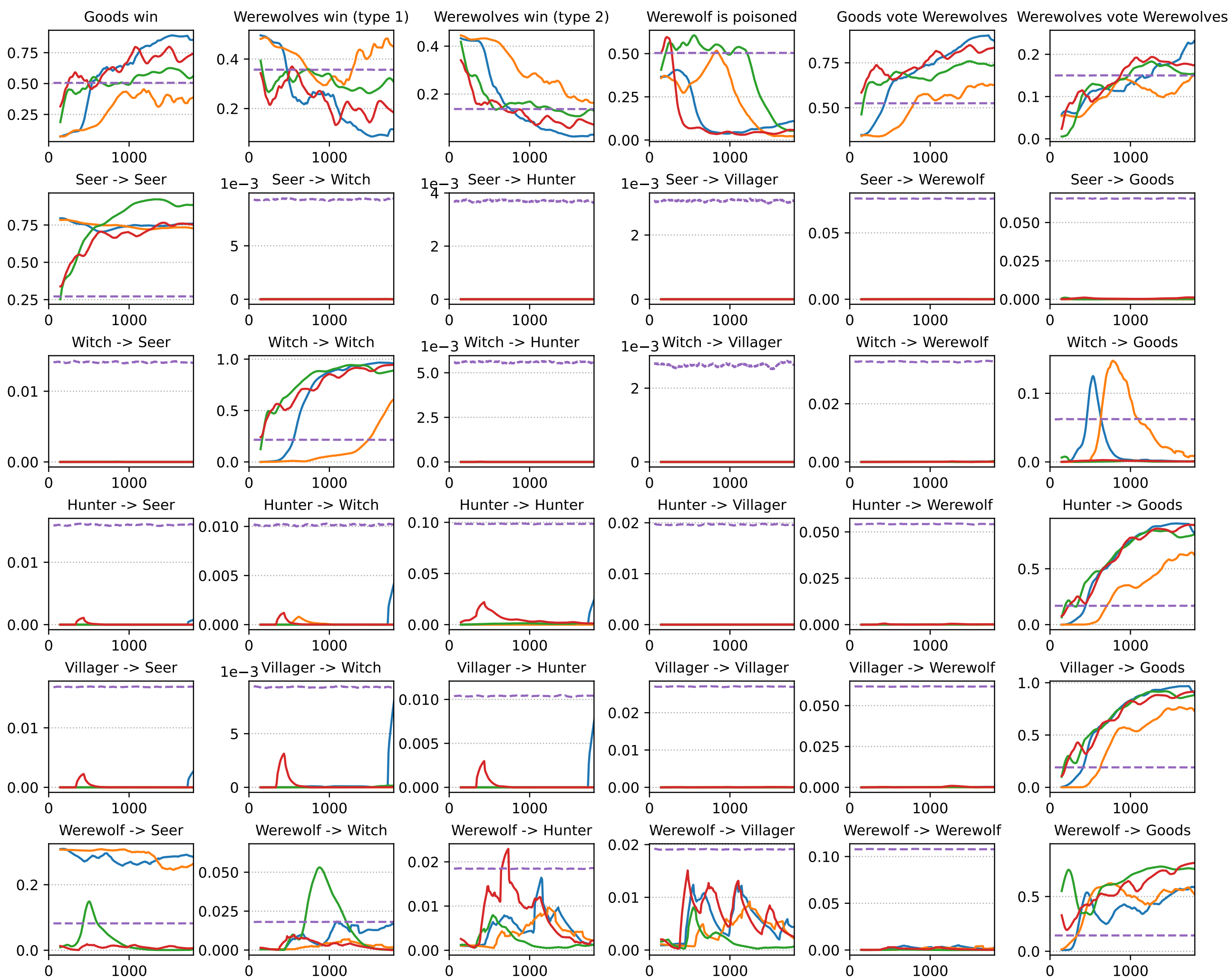


Figure 6: Detailed training curves for different agents during RL training. The x-axis represents the training steps (k), and the y-axis represents the probability. The horizontal line in each subplot corresponds to the probability observed in human data. "Werewolf -> Seer" represents that a Werewolf claims that he is the Seer in the speech.

Seer: The Seer can verify a player's faction each night (a Werewolf or the "Good"), but cannot know their specific role. The Seer cannot verify himself or any player who has already been verified.

Witch: The Witch possesses an antidote and a poison. The antidote can save a player killed by Werewolves at night, and the poison can kill a player. The Witch cannot use both potions in the same night and can only save herself on the first night.

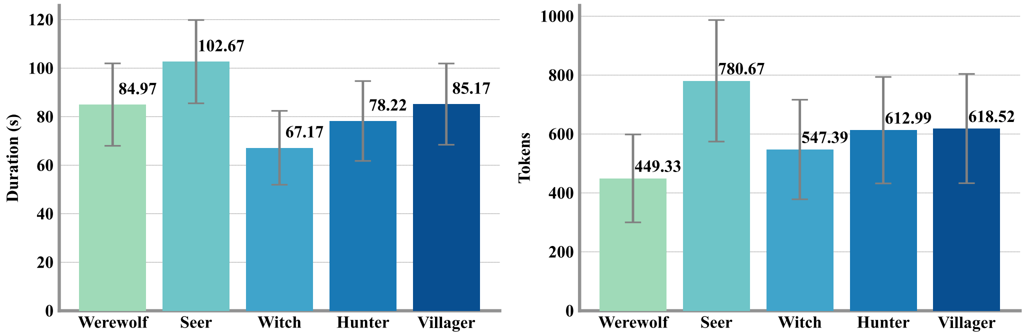
Hunter: When the Hunter is killed by Werewolves at night or voted out during the day, he can shoot a player. However, the Hunter cannot use his ability when poisoned by the Witch.

C.3. Game Task Flow

The game proceeds in a night-day cycle until the victory conditions are met.

The night tasks flow:

(1) Werewolves decide to kill a player. In our simulation of the game environment, we have simplified the discussion into a three-round voting process. During voting, werewolf players can see their teammates' previous votes.



(a) Speak duration of roles **(b) Speak tokens of roles**

Figure 7: Speech duration and token length categorized by roles in *FanLang-9* dataset.

(2) The Witch uses her ability.
(3) The Seer uses his ability.

The daytime tasks flow:

(1) The Moderator announces the deaths from last night but does not reveal the causes of death.
(2) Deceased players give their last words (only for the first day).
(3) If deceased players have additional abilities, they may choose to use them.
(4) First round of speeches. The speech sequence is determined by the following rules: (a) if no player died last night, randomly select an initial speaker and randomly decide a clockwise or counterclockwise speaking order. (b) randomly select a deceased player and start the speaking order clockwise or counterclockwise from him. Players cannot interrupt others during their speeches.
(5) First round of voting. Each player votes for a single player to exile from the game. Other players' voting choices remain hidden until the voting session ends.
(6) Second round of speeches. If there is a tie in the first round of voting, the tied players give their second speeches; otherwise, the process moves on to task (8) The first speaker, selected randomly from the tied players, initiates the sequence, which could proceed either clockwise or counterclockwise.
(7) Second round of voting. If there is still a tie after the second vote, the game moves on to the next night, and no player is exiled.
(8) The exiled player gives his last words.
(9) If exiled players have additional abilities, they may choose to use them.

D. Analysis of the *FanLang-9* Dataset

The *FanLang-9* dataset consists of 18,800 recordings, 260K speech instances, with an average speech length of 500 characters. Specifically, the following characteristics underscore the unique nature of the dataset:

D.1. Speech Duration and Length

Figure 7 (a) demonstrates significant variations in speech duration among different roles, with an average of approximately 90 seconds each. The Seer's inspection information at night forms the core and fundamental logical basis of the game. Therefore, it is the Seer's duty to share inspection information, provide persuasive speeches, and lead discussions during the speech phase, resulting in the longest duration among all roles. Besides, Werewolves and Villagers need to convincingly identify themselves and predict the roles of other players, necessitating detailed and logical analysis. In Figure 7 (b), the dataset shows the shortest token length among Werewolves, which is not correlated with their speaking time. This suggests that Werewolves' speeches are relatively concise, which may stem from the complexity of deception that requires more time to strategize. We further illustrate the distribution of token length in a single speech in Figure 8.

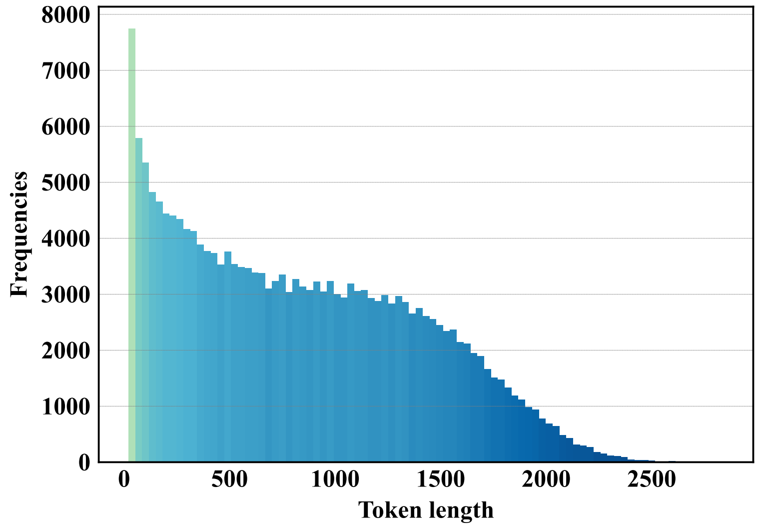


Figure 8: Distribution of speech token length.

D.2. Tokenization and Categorization of Speeches

The reasoning result of a speech produced by the *Listener* is formatted in JSON style, containing pairs of player ids with their attributes. The result typically includes phrases and word groups containing multiple attributes, probabilities, and irrelevant information, e.g., "seems to be a werewolf: [3, 6]", "cannot hear clearly: [8]". We then tokenize and categorize the result into related identities and actions, along with their probabilities, as shown in Table 7. The final language features account for 96.09% of the *FanLang-9* dataset, capturing the majority of the information expressed by speakers in the Werewolf game.

D.3. Voting Preference

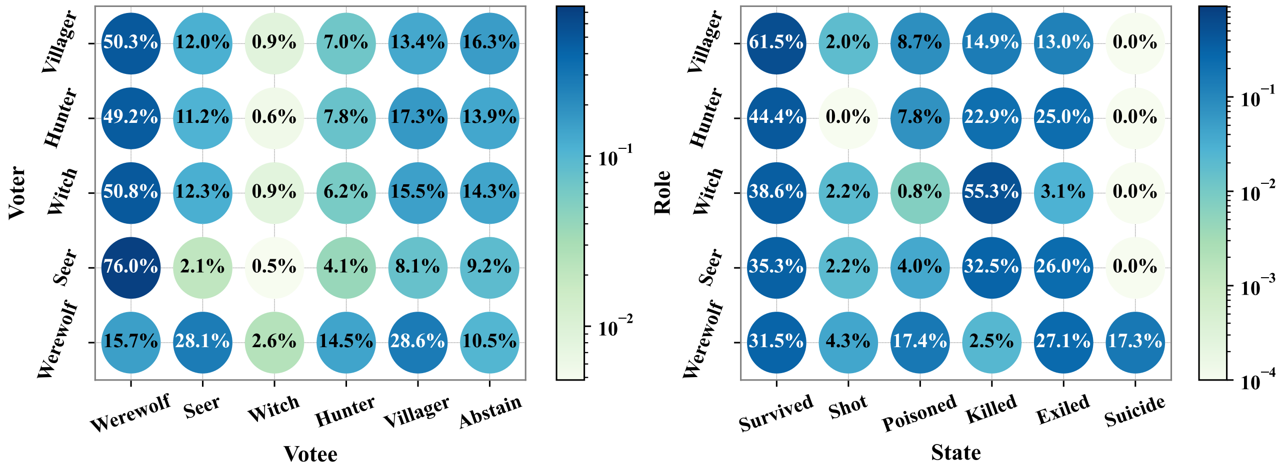
We analyze how human players tend to vote in the perspective of different roles in Figure 9 (a). As for voting werewolves, the Seer has the highest accuracy of voting Werewolves due to his inspection ability, while Werewolves vote for their teammates with a probability of 15.7%, aiming to disguise themselves as the "Good" faction. The other roles have a 50% chance of voting for Werewolves, since they lack additional information beyond the game state and historical speeches. As for voting from Werewolves, the most prioritized target are the Villagers (28.6%), since they have the least amount of information and are easier to be incriminated as Werewolves. The second prioritized target is the Seer (28.1%), since the Seer can inspect players' identities, it is crucial to remove him out of the game as soon as possible.

D.4. Final State of the Roles

In Figure 9 (b), we present the final states of roles in the end of the game, categorized as *Survived*, *Shot* by the Hunter, *Poisoned* by the Witch, *Killed* by Werewolves, *Exiled* after the Voting stage, and Werewolves committed *Suicide*. Notably, the Witch has the highest likelihood of being killed by the werewolf at night (55.3%), with the seer following at 32.5%. Werewolves commit suicide with a probability of 17.3%, and are killed by their teammates at night with a probability of 2.5%. During the daytime voting, Werewolves are the most frequently exiled role, indicating their challenges in providing deceptive statements, while the Witch has the lowest probability, reflecting their effectiveness in gaining trust through speeches.

D.5. Win Rate

Table 6 illustrates that in human gameplay, the win rates for the Good and Werewolf factions are closely matched.



(a) Role voting in the Werewolf game **(b) Final state of roles**

Figure 9: (a) The voting probability distributions for players with different identities in all voting sessions; (b) the final survival status and causes of death probabilities for players at the game's end.

Table 6: Win rate in the *FanLang-9* dataset.

Camp	Win number	Win rate
Goods	9293	49.31%
Werewelf	9554	50.69%

E. Ethical Considerations

With the integration of LLMs into complex reasoning tasks, such as those demonstrated in social deduction games like Werewolf, we are witnessing the emergence of AI agents that not only mimic human-like reasoning but also engage in communications that could be considered deceptive by nature. While these developments showcase the potential of AI to understand and navigate intricate human interactions, they also raise important ethical and societal considerations that must be addressed. To address these ethical and societal challenges, we propose several mitigation strategies:

Transparent communication and monitoring: Our framework ensures transparency through explicit structured information at every stage of the AI's decision-making process, from listening and reasoning to speech generation. To enhance this transparency, we propose implementing real-time transparency logs that capture and display the reasoning paths, identity predictions, and speech instructions generated by the AI. By having a complete audit trail, we can monitor the AI's decision processes, ensure adherence to ethical guidelines, and trace any unintended actions back to their source.

Control and filtering mechanisms: Our speech instructions are enriched with contextual information specific to the Werewolf game, allowing for robust control over the fine-tuned LLM. To further mitigate potential negative impacts, we propose implementing dynamic contextual guardrails. These guardrails will utilize our existing filtering mechanism (as outlined in Section 3.4) to not only match generated speech with instructions but also to check against a set of ethical and societal norms. If the AI's output is flagged as potentially harmful or deceptive beyond the game's scope, it will be withheld and replaced with a template response. This additional layer of control will act as a safeguard against the misuse of AI in generating deceptive or manipulative content outside the intended gaming environment.

F. Implementation Detail

F.1. Evaluation Criteria for the Speech Generation

The human evaluation requirements for speech generation are as follows.

Table 7: Tokenization and categorization of speeches on the *FanLang-9* dataset.

Tokenized attributes	Is	Might be	Is not	Might not be	Is not sure	Ratio	Accumulation
Werewolf	178,423	27,297	516	313	15	26.55%	26.55%
Good (the good faction)	83,071	622	85	73	10	10.77%	37.32%
Vote	68,853	87	81	1	3	8.87%	46.19%
Seer	60,339	114	111	321	8	7.82%	54.01%
Witch	35,408	42	29	8	3	4.56%	58.57%
Gold Water (checked Good)	34,727	8	8	1	/	4.46%	63.03%
Check (Seer's inspection)	26,027	17	17	/	/	3.35%	66.38%
Poison	21,897	82	9	1	/	2.83%	69.21%
Villager	21,611	28	19	10	1	2.78%	71.99%
Werewolves' target	19,481	17	12	/	1	2.51%	74.50%
Hunter	17,603	26	70	5	2	2.28%	76.78%
Silver Water (saved)	14,016	3	5	1	2	1.80%	78.58%
Suicide	3,826	4	1	/	1	0.49%	79.07%
Uncertain Identity	/	/	/	/	2,937	0.38%	79.45%
Shoot	1,100	2	2	/	/	0.14%	79.59%
Save (by the Witch)	1,065	/	/	/	/	0.14%	79.73%
Abstain voting	683	3	1	/	/	0.09%	79.82%
Special Role	273	4	/	/	/	0.04%	79.86%
Irrelevant Information	126,279	/	/	/	/	16.23%	96.09%
Unprocessed	30,476	/	/	/	/	3.91%	100.00%

Legality: Absence of obvious logical errors and illegal statements that are conflicted with game rules, such as:

- "I am a Werewolf."
- "I am the Seer, and I poisoned Player 5 last night."
- "Player 3 is a good person; I suggest voting for him."
- "I suggest voting for myself."
- "Player 8 is a Werewolf, he was voted out and took Player 6." (Player 8 is the hunter and publicly shot Player 6).
- "I suggest voting for Player 8." (Player 8 has already been voted out).

Reasonableness: of the speeches, such as

- The Seer correctly reports his inspection last night.
- Werewolves reasonably disguise their identity, employing various strategies such as pretending to be the Seer, aggressive claims, and betraying their teammates.
- Villagers make reasonable guesses about the Good faction and Werewolves.
- Note: the correctness of guessing other players' identities is not part of the evaluation criteria.

Other: factors unrelated to key information:

- Language style, colloquial expression, game jargon.
- Presence of verbose or redundant statements, such as greetings or defending the village community.

The evaluation criteria are in descending order of priority. For example, if model A has no obvious logical errors but its speech is not very reasonable, and model B has obvious logical errors, then A is better than B. For the ranking of the five samples, if there are obvious logical errors, mark them as -1 and no need to rank them. For example, if models A and B have obvious errors, the annotation result could be: $\mathrm{A}:-1,\mathrm{B}:-1,\mathrm{C}:1,\mathrm{D}:2,\mathrm{E}:3$, where 1 represents the best and 5 represents the worst. Apart from marking illegal statements as -1, tied rankings are not allowed.

Table 8: Behavior scores applied in the 9-player werewolf game.

Role	Description	Score
Seer	If a werewolf is exiled in the first day	$+0.5$
	For giving up the inspection at night	-0.5
Witch	For poisoning a werewolf	$+1.0$
	For poisoning a good player	-1.0
Hunter	For shooting a werewolf	$+1.0$
	For shooting a good player	-1.0
Good roles except the Seer	For voting for a werewolf	$+0.5$
	For voting for a good player	-0.5

F.2. *Thinker* Model Structure

The architecture of the *Thinker* network is designed to capture the intricacies of gameplay from the perspective of the current player, which encompasses speeches, actions, and game status information of all nine players involved, including themselves. We employ a shared-parameter feature encoding network that processes the data for each of the nine players individually.

For the i-th player, up to 10 language features $\mathbf{F}$ are stored. These language features are enriched with headers indicating the time-tag, type, and order of the speeches. Subsequently, these annotated language features are processed through another shared-parameter speech feature encoding network, which consists of a three-layer (181-256-256) multilayer perceptron network (MLP). After processing the ten pieces of features, a *reduce_mean* operation is applied to the outputs to synthesize the overall speech embedding for the player e_i^{speech}. This synthesized speech embedding is then combined with additional game state information such as the player's actions, status, and other relevant data. The aggregated data is fed through a feature encoding network (again, a three-layer MLP of 1019-512-512) to generate the feature embedding for the i-th player e_i.

In the final step, the feature embeddings of all nine players $e_1, e_2, ..., e_9$ are subjected to a *reduce_mean* operation to create a collective feature encoding. This comprehensive encoding is then passed through an all-players feature encoding network (a three-layer MLP of 523-512-512) to construct the corresponding action decision, identity prediction headers, as well as speech instructions.

F.3. Reward Shaping

Drawing inspiration from the concept of the Behavior Score, we have devised the reward shaping for *Thinker* in the reinforcement learning to circumvent illegal actions and speech that may arise during unfettered exploration within the AI Werewolf game. The specifics of this mechanism are outlined in Table 9. It encompasses several key areas:

- Game result reward: The AI receives a reward based on the win or loss, survival duration at the end of a game.
- Action reward: for taking actions that are deemed appropriate and effective within the context of the game.
- Speech reward: to incentive the AI to engage in communication that is beneficial to its goals, such as persuading other players or disseminating useful information.
- Action-Speech consistency reward: to stimulate coherence between what the AI says and does, a reward is given for alignment between the AI's declared intentions in speech and its subsequent actions.
- Cognitive reward for Werewolves: Central to the training of a Werewolf AI is the ability to masquerade as a member of the "Good" faction. To enhance this capability, we provide a reward based on the change in identity prediction from the perspective of the "Good" players. The better a Werewolf AI can deceive the "Good" faction about its true identity, the larger the reward it receives.

F.4. Details of Overall Training Process

We provide Pseudo-code in Algorithm 1, the *Thinker* and LLMs are trained separately in our framework. This design choice was intentional and serves as one of the strengths of our framework. The separation facilitates training efficiency since LLMs, which we employ as both *Listener* and *Presenter*, are inherently slower in sample generation compared to the *Thinker* module. Therefore, to optimize our training process, we either employ offline RL or decouple the training of the *Thinker* and LLMs. The inference workflow is as follows: *Listener* (LLM) -> language feature **F** -> *Thinker* (RL) -> speech instruction **I** -> *Presenter* (LLM)

During the training of the *Thinker*, the generated speech instructions **I** are treated as the new input language features **F** for the subsequent steps, allowing for a seamless integration of the RL training into the overall process. Our hybrid training framework incorporates both BC and PPO. During training, each game session has a certain probability of being a BC or RL game. In a BC session, actions a and speaking instructions **I** are taken directly from human replay, bypassing the *Thinker* inference. Conversely, in an RL session, the *Thinker* actively generates actions and speaking instructions. Samples from the game session are tagged as either BC or RL. For the Learner, BC samples utilize the BC loss mentioned in Equation 2, while RL samples employ the PPO loss Equation 3.

F.5. Training Hyper-parameters

The training hyper-parameters for the *Thinker* are provided in Table 10.

Regarding the hyperparameters in Equation 4: The Behavioral Cloning coefficient α determines the extent to which the RL policy refers to human strategies versus greedily selecting the RL strategy. We observed that when α decays to 0, werewolves completely abandon the strategy of claiming to be the Seer, because the difficulty for werewolves to pretend to be the Seer is high, and it is relatively challenging for RL to optimize. A more favorable choice is to masquerade as a villager. Therefore, we still maintain a small $\alpha = 0.01$ during the later stages of training. As for the auxiliary task coefficient β, we tested values in $\{1.0, 0.1, 0.01\}$, and found that they had minimal impact on RL, as it is an auxiliary learning task.

The fine-tuning hyper-parameters for the *Listener* and *Presenter* are provided in Table 11.

F.6. LLM Prompting for *Listener* and *Presenter*

The information extraction prompt for the *Listener* module contains the following parts:

- Description of the background of the Werewolf game, as shown in Table 12, which provides the game configuration, game rules, terminology, and descriptions of roles' identities and skills.
- Task requirements, as shown in Table 13. The prompt describes the structured information in JSON format that we expect LLMs to produce, and we describe the appropriate values for each position of the structured command and limit the output within a reasonable range.
- Few-Shot examples, as in Table 14, which provides examples of correctly extracted information from the speeches of different identities and skills, to improve the accuracy of the task as well as to align it with the type of output we expect.
- Current information: Finally, we input the current speech of the player, the game state, e.g., the speaker's *Player id*, role, the current speech types, as in Table 15, to prompt LLMs for deductive reasoning.

The speech generation prompt for the *Presenter* module contains the following parts, as shown in Table 16:

- Description of the background of the Werewolf game, which is the same as in the *Listener* module.
- (Optional) speech instruction. The prompt is a structured output from the *Thinker* module, and its meaning aligns with that of the *Listener* module, with a 1-shot example.
- Task requirements, which is the similar to that in the *Listener* module expect for the speech generation task.
- Current information, which is the similar to that in the *Listener* module except that we prompt all the historical speeches.

F.7. Game Log Examples

Table 17 presents a comprehensive analysis of a 9-player werewolf game log, culminating in a victory for the Werewolf.

Algorithm 1 Pseudo-code for the overall training process.

Require:

- Data pairs 1: for finetuning of the *Listener*
  Input: [game state s, historical speeches $\mathcal{H}$, current player's speech $\mathbf{S}$]
  Output: [language feature $\mathbf{F}$]
- Data pairs 2: for finetuning of the *Presenter*
  Input: [game state s, historical speeches $\mathcal{H}$, speech instruction $\mathbf{I}$]
  Output: [current player's speech $\mathbf{S}$]
- Data pairs 3: for behavioral cloning of the *Thinker*
  Input: [game state s, historical collection of all language features $\mathcal{F}$]
  Output: [action a], or [speech instruction $\mathbf{I}$], decided by the current task type.

Listener* and *Presenter:
if *use APIs* **then**
  Listener: Use API for generating language features $\mathbf{F}$.
  Presenter: Use API for generating speeches $\mathbf{S}$.
else
  Listener: Finetune model with Data pairs 1 and hyperparameters in Table 11.
  Presenter: Finetune model with Data pairs 2 and hyperparameters in Table 11.

Thinker:
Initialize network parameters for a population of P agents: $\{\theta_1, \theta_2, ..., \theta_P\}$.
Start multiple actors and learners in parallel.
Actors: **while** *true* **do**
  Fetch the latest model from the learners. Add the latest checkpoint into a checkpoint list.
  Sample $N-1$ checkpoints from the list and the latest checkpoint.
  Decide the game episode is BC or RL, run an N-player game episode.
  if *game episode is BC* **then**
    Get behavioral cloning training samples from Data pairs 3.
  else
    Generate RL training samples.
  Accumulate samples in the form $x = (s, \mathcal{F}, a, \mathbf{I}, r, \text{is\_BC})$ and send them to the replay buffer.
Learners:
for $t \in 1, 2, 3, ...$ **do**
  for $p \in 1, 2, ..., P$ **do**
    Fetch a batch of samples for agent p from the replay buffer.
    Calculate value loss and policy loss according to PPO algorithm in Equation 3.
    Calculate behavioral cloning loss according to Equation 2.
    Calculate loss for auxiliary tasks.
    Update parameters θ_p using gradients on loss in Equation 4.

Table 9: Reward shaping in the RL training of the *Thinker*.

Description	Reward
# Game reward	
the Good faction win, Werewolves get	-4
the Good faction win, Villagers and special roles get	$+2$
Werewolves win, Werewolves get	$+4$
Werewolves win, Villagers and special roles get	-2
Any player survives for a new day	$+1$
# Action reward	
the Goods vote for a Werewolf	$+2$
the Goods vote for a Good role	-2
the Witch poisons a Werewolf	$+2$
the Witch poisons a Good role	-4
the Hunter shoots a Werewolf	$+2$
the Hunter shoots a Good role	-4
# Speak reward	
the Seer claims his identity	$+2$
the Witch claims his identity	$+1$
the Goods correctly identify a Werewolf in the speech	$+2$
the Goods wrongly identify a Werewolf in the speech	-2
the Goods correctly identify a Good role in the speech	$+1$
the Goods wrongly identify a Good role in the speech	-1
Any player who claims that he is a Good role	$+0.5$
# Action-Speech correlated reward	
the Seer correctly share his inspection last night	$+2$
the Witch correctly share the usage of antidote or poison	$+1$
any player who claims the voting intention and then vote the same player	$+1$
# Cognition reward	
the change δ of summation of a Werewolf's identity probabilities in the Goods' perspective:	
as the Seer	4δ
as the Witch	2δ
as the Hunter or Villagers	1δ

Table 10: Hyperparameters for the *Thinker* training.

Hyperparameters	Value
Population size	4
Number of actors	700 (CPUs)
Number of learners	8 (GPUs)
Replay buffer size	100k
Mini-batch size	2048
Optimizer	Adam
Learning rate	2e-4
Discount factor (γ)	1.0
GAE parameter (λ)	0.9
PPO clipping ratio	0.2
Value function coefficient c_1	0.5
Entropy coefficient c_2	0.05
Behavioral Cloning coefficient α	$0.1 \rightarrow 0.01$
Auxiliary task coefficient β	0.1

Table 11: Hyperparameters for fine-tuning the *Listener* and *Presenter*.

Parameter	*Listener*	*Presenter*
# Basic Training Parameters		
Learning rate	1e-4	1e-4
Sequence length	4096	8192
Optimizer	AdamW	AdamW
Adam beta1	0.9	0.9
Adam beta2	0.999	0.999
Adam epsilon	1e-8	1e-8
Train batch size	32	8
Train epochs	3	3
Max steps	5000	10000
Warmup steps	500	1000
Max grad norm	1.0	1.0
# Model Configuration		
Hidden size	4096	
KV channels	128	
Num layers	28	
Num attention heads	32	
Layer norm epsilon	1e-5	
Torch dtype	float16	
# Distributed Training Settings		
TP size	2	
PP size	1	
# Attention Mechanism Configuration		
Multi query attention	True	
Multi query group num	2	

```
Werewolf Game Background Prompt

# Task Scenario: 9-player Werewolf game speech.

"Good" Faction:
- 3 Villagers
- 1 Seer
- 1 Witch
- 1 Hunter

Werewolf Faction:
- 3 Werewolves

Common terminologies are explained as follows:
1. Werewolf, bandit, wolf, bad faction, knife: Werewolf.
2. Villager, civilian, white card: Villager.
3. Seer, prophet: Seer.
4. Witch, witch card: Witch.
5. Hunter, gun: Hunter.
6. Gold, gold water, verified Good: A good person verified by the Seer.
7. Verify Kill: A Werewolf verified by the Seer.
8. Silver, silver water, Werewolves' target, Saved: A person saved by the Witch.
9. Iron, steel, certain: Very certain, e.g., "Player 3 is an iron Werewolf"
or "Player 3 is definitely the Werewolf," indicates that Player 3 is certainly a
Werewolf.
10. Jump: A player declares his/her role (not necessarily his/her true role).
11. Backstab: A Werewolf sides with the good people, betraying their own
teammates.
12. Defame: To demean the identity of other players.
13. Exalt: To believe in the identity of other players.
14. Vote out, point, nominate, ballot: Voting, e.g., "Vote for Player 6 or Player
7," means to vote Player 6 or Player 7 out.
```

Table 12: Werewolf game background prompt.

Speech Understanding Requirements Prompt

 # Task requirements are as follows:

Based on your understanding of the game state and speeches, please output the extraction results in JSON format in sequence. The format should be:

```
{
    "identities": {"<identity>": [player,player,...]} ,
    "actions": {"<action>": [subject player -> object player,
                subject player -> object player]}
}
```

Example:

```
{
    "identities": {"werewolf":[3,5]}, {"<action>": [subject player -> object player,
                    subject player -> object player]}, }
    "actions": {"check":[1->6, 2->3]}
}
```

- This indicates Players 3 and 5 are werewolves, Player 1 checks Player 6, and Player 2 checks Player 3.
- Player numbers can only be: 1, 2, 3, 4, 5, 6, 7, 8, 9.
- When players express their intentions, please correspond to the identity of the player, for example, if Player 5 speaks, then consider from the perspective of Player 5.
- The subject number should be inferred from the context, such as 'I', 'you', 'he', 'she', etc. If unknown, use 'unknown', for example: "check":[unknown->6].

Possible JSON KEYs are:

Identities:
- Roles: Seer, Witch, Hunter, Villager, Werewolf, "Good" faction, Werewolf faction, gold water, silver water, the werewolves' target, etc.
- Guess: suspicious, credible, uncertain, tolerant, etc.
- Speech: good (up), bad (down), listen well, listen to kill, etc.
- Faction: allied, support, werewolf candidate, etc.
- Online status: disconnected, offline, not online, voice, etc.

Actions:
  - Skills:
    - Seer: check, inspect.
    - Witch: poison, save.
    - Hunter: shoot, take away, crash, kill.
    - Werewolf: self-destruct, explode.
  - And skills that will be used in the future:
    - Vote: vote out, choose a target, etc.

- Quotes from other Players' statements do not need to be summarized.
- Note the distinction between quantifiers and player numbers: must be, that there are three werewolves.
- Note negative statements: not, impossible, implausible, not quite, etc.
- Note the abbreviation of number + information, e.g., "three golds, nine slashes, one, six, eight, three wolves" results in:"identities": "gold water":[3], "slash":[9], "werewolf":[1,6,8]

Table 13: Speech understanding requirements prompt.

Information Extraction Few-Shot Prompt

 # The following are 11 speeches and corresponding information extraction examples:

Player 3 spoke: "I checked Player 6, and I suggest Player 8 turn around and vote for Player 6. I will check the identity of Player 4 in the next round."

```
{
    "identities":{"seer":[3],"werewolf":[1,6,8]},
    "actions":{"check":[3->6],"suggest to vote":[8->6],
               "check in the next round":[3->4]}
}
```

Player 7 spoke: "Player 2 and I are collaboratively searching for a Seer. Player 2 assists the good faction in combating werewolves. There's a possibility that Player 9 is a werewolf, although I am not certain. The behavior of Player 9 seems suspiciously similar to that of Player 2, who possesses the ability to shoot. Additionally, Player 4 is identified as a Witch. Regarding the usage of silver water, I suggest targeting Player 6."

```
{
    "identities":{"maybe a wolf":[9],"hunter":[2],"silver water":[4]},
    "actions":{"suggest to vote":[7->6]}
}
```

Player 9 spoke: "Player 8 is the gold water. Player 2 is not a werewolf, neither is Player 3. However, Player 7 is suspicious, and I recommend voting against Player 7. The roles of Player 4 and Player 5 are unclear, and Player 1 suspects both of them to be werewolves. I advise Player 7 to use poison, which could help confirm my role as a Seer. Concerning the hunter, there is a standoff between Player 8 and myself. If there is any uncertainty about Players 1, 2, or 4, the gun should be used in this situation against Player 2. Now, it's time for Players 4 and 7 to present their arguments, and there is no need to focus on Player 9."

```
{
    "identities":{"gold water":[8],"good camp":[2,3],"suspicious":[7],
                  "werewolf":[4,5],"seer":[9] ,"werewolf candidate":[1,2,4],
                  "hunter":[2],"debate players":[4,7]},
    "actions":{"suggest to vote":[9->7],"suggest to poison":[unknown->7]}
}
```

Player 3 spoke: "Being the first player to speak, my turn was conveniently arranged. However, I am uncertain about Player 2's allegiance. In my view, Player 2 lacks credibility."

```
{
    "identities":{"no result": []},
    "actions":{"no result": []}
}
```

Player 7 spoke: "Player 3 will be poisoned tonight. I hold the Witch card. I heed the guidance of the two players with gold cards. Players 9 and 5 are identified as wolves. Players 4 and 6 hold cards corresponding to their numbers, with Player 4 being more trustworthy than Player 5. Player 3 cannot be revived. To preserve my own safety, I will reveal myself as the Witch. I have already used the silver water card on Player 1. Player 9 remarked that I should be pleased with this misfortune, indicating that the prime werewolf card was passed to a fellow teammate."

```
{
    "identities":{"witch":[7],"gold water":[2],"werewolf":[9,5],"suspicious":[4]},
    "actions":{"suggest to poison":[7->3],"believe to be a silver water":[7->1]}
}
```

Player 8 spoke: "Player 5 appears highly suspicious. He could either be a werewolf or might be deceiving his teammates. His failure to set wolf traps, dishonesty about the wheat sequence, and excessive talking during the first microphone turn is concerning. Players 6 and 7 might be superficial wolves. Player 7, however, seems to have a sensible perspective and could be part of the good camp. I recommend voting against Player 5."

```
{
    "identities":{"suspicious":[5],"werewolf":[6,7],"good camp":[7]},
    "actions":{"suggest to vote":[8->5]}
}
```

Player 2 spoke: "Regarding the game, my suspicion falls on Players 1, 5, 7, and 3 as potential wolves. The accusation by Player 3, however, is incorrect. I find Player 3's judgment flawed. It's frustrating. Similarly, I suspect that Players 1, 5, 7, and 3 are wolves according to Player 5's perspective. Let's test this theory. I propose we eliminate Player 5 today, and then I, as a Witch, will poison Player 7 tomorrow night. Observe the game's progression tomorrow, and you will see that both Player 5 and I, as Witches, agree on Player 2, and our views align with Player 3's decision. Therefore, I request that we focus on Player 5 first."

```
{
    "identities":{"werewolves' target":[3],"werewolf":[1,5,7],"witch":[2]},
    "actions":{"suggest to vote":[2->5, 2->7]}
}
```

Player 1 spoke: "Player 6 is engaging in killing actions. Players 5 and 7 have been poisoned. Players 4 and 5 are both targeting Player 1. Player 3 has been stabbed, and it's possible that Players 2, 4, and 9 each represent a threat, akin to three knives. Player 5 has revealed themselves as the Witch and has provided Player 3 with a dose of silver water."

```
{
    "identities":{"seer":[1],"poison":[5,7],"depreciate":[4,5],
                  "werewolves' target":[3],"werewolf":[2, 4,9],"witch":[5]},
    "actions":{"check":[1->6],"believe to be a silver water":[5->3]}
}
```

Player 1 spoke: "I, Player 1, am part of the good faction. The focus of today's game is on Players 3 and 5. Player 9 might be a werewolf. I did not use any poison last night."

```
{
    "identities":{"good camp":[1],"werewolf":[9]},
    "actions":{"suggest to vote":[1->3,1->5]}
}
```

Player 9 spoke: "I am the Hunter. Player 7 has self-destructed. Player 2 might be associated with the silver water. As for myself, I reiterate that I am the Hunter. Player 1 is acting suspiciously, resembling a white card. I request the Witch to acknowledge this. Player 3 is overly concerned with external cards, which is uncharacteristic of a Prophet. Players 3 and 8, please return to the game, as there's still an opportunity for a round of confrontation."

```
{
    "identities":{"hunter":[9],"self-destruction":[7],"silver water and seer":[2],
                  "white":[1],"not like a seer":[3] },
    "actions":{"suggest to vote":[9->3,9->8]}
}
```

Player 4 spoke: "I believe Player 6 is trustworthy as he revealed Player 6's key card. My intention is to verify Player 3. Player 7, who holds the gold water, should cast their vote against Player 8. It's evident that Players 3 and 7 are not the same individual. On the field, there are only two players acting as villagers. I have identified the three wolves. There is no necessity to doubt Player 7; instead, Player 4 can be acknowledged as the Seer."

```
{
    "identities":{"gold water":[7],"seer":[4]},
    "actions":{"consider credible":[4->6],"verified":[4->3],
               "suggest to vote":[4->8]}
}
```

Table 14: Information extraction few-shot prompt.

```
LLM prompting for the Listener

# Task type:  Information Extraction

${{ Werewolf Game Background Prompt }}
${{ Speech Understanding Requirements Prompt }}
${{ Information Extraction Few-Shot Prompt }}
# The task text is as follows:

Player 8 spoke:  "I think Player 9 is a good person, but I am not sure about the
identities of Player 5 and Player 6."

Please directly output the information extraction result in JSON format:
```

Table 15: LLM prompting for the *Listener*.

Speech Generation Prompt

Now that you play as a Werewolf player, I'm going to provide you with some information about the position you're about to speak in, which hasn't happened yet and is not historical information, and ask you to concatenate this information to generate a paragraph of speech text.

First, I'll give you some background on the game:

 # Task type: Game Dialog Generation

${{ Werewolf Game Background Prompt }}
You are playing a 9-player werewolf game. Suppose you're game Player 1, and your identity is Seer.

I provide you with the format of the in-field message:

```
{
    "identities": {"<identity>": [player,player,...]} ,
    "actions": {"<action>": [[subject player, object player],
                [subject player, object player]]}
}
```

Example:

```
{
    "identities": {"werewolf":[3,5]}
    "actions": {"check":[[1,6],[2,3]]}
}
```

- Indicate that Player 3 and Player 5 are werewolves, Player 1 checks Player 6, Player 2 checks Player 3, and the subject and object are irreversible.
- The only possible player IDs are 1,2,3,4,5,6,7,8,9, and unknown should be replaced by the speaker's player ID.

Note that the generated speech result should strictly fulfill the following 10 requirements:
1. Include all the information in the information extraction result.
2. Don't over-imagine and introduce hallucination, and prioritize the accuracy of the information.
3. The logic between the generated results should be in line with the position of the players in Wolfsbane, and there should not be any contradictions between the logic before and after.
4. Pay attention to the diversity of generated results.
5. The generated results should be as anthropomorphic as possible, imitating the speaking style of human players.
6. Please be firm in your belief that you are the Good faction, whether you yourself are in the Good faction or the Werewolf faction.
7. Identities or actions can be left out if the result is empty, empty is invalid information.
8. A player can only be one of the roles of Villager, Seer, Witch, Hunter, or Werewolf, for example, it's impossible to be a Witch and a Hunter at the same time, if there is more than one conflicting Werewolf identity in the information I've provided you with, please randomly choose one.
9. Please state your identity first.
10. Do not make statements such as "unite", "stay alert", "defend the village", "together we will go to victory", "find out the werewolf as soon as possible", "keep watching", "the information shows up", "hopefully we will find it", "think about what we should do next", "keep discussing" or similar statements.

```
Now that you are Werewolf Player 1, with the identity of Seer, I am providing you
with the key information for your upcoming statement:

{
    "identities": {"villager": [1]},
    "actions": {"no result": []}
}

Please follow the key messages to the letter and keep the text under 100 words:

{
    "identities": {"villager": [1]},
    "actions": {"no result": []}
}

Now it's your turn to speak:
```

Table 16: Speech generation prompt.

Werewolf Game Log Example

game role assignment:
- Player 1: Werewolf.
- Player 2: Witch.
- Player 3: Villager.
- Player 4: Werewolf.
- Player 5: Villager.
- Player 6: Hunter.
- Player 7: Villager.
- Player 8: Werewolf.
- Player 9: Werewolf.

day 1, night:
- Werewolves: Player 4, Player 8, and Player 9 choose to kill Player 5.
- Seer: Player 1 investigates Player 7, who is in the good camp.
- Witch: the werewolf's target is Player 5, and the witch chooses to use a potion to save Player 5.

day 1, announcement:
- no players were out last night.
- remaining players: Player 1(Werewolf), Player 2(Witch), Player 3(Villager), Player 4(Werewolf), Player 5(Villager), Player 6(Hunter), Player 7(Villager), Player 8(Werewolf), Player 9(Werewolf).

day 1, discussion:
- (random, Player 9 starts to speak)
- day 1 Player 9 (Werewolf) said: "Good man card, ah, the first position to speak is not my specialty, you say the system and gave me this right, so that I can openly paddling, hey, I especially want to exercise, I will say Player 2, just more than a sentence do not say it, OK? On the first day to find the next day we in that what in the look at the side of the station to update the speech, okay? Player 9 is a white ah, Player 9 Player 8 license plate"
- day 1 Player 1 (Seer) said: "The Player 7 is gold water, with a sense of hearing, to lead the good guys to victory. My last words are to keep an eye on Player 8 and Player 9 as they are close friends."
- day 1 Player 2 (Witch) said: "Player 7 has an average sense of hearing and cannot take sides. Player 1 listens to the situation and listens to the speech. Player 7 only listens to Player 8 or Player 9, do the other cards enter the perspective? Player 5 is a silver water cutter, I saved him, don't hold a grudge. Before Player 6, Player7 and Player 8 werewolves should not be less, can go to the highest identity."
- day 1 Player 3 (Villager) said: "I think Player 4 need be intimidated, Player 7 has no sincerity, Player 9's speech is error tolerant, Player 2 keeps taking the Witch card, and he's good."
- day 1 Player 4 (Werewolf) said: "I'll give gold water on Player 2, and Player 9 is labeled as a werewolf. Player 7 is a predecessor of the Player 5 and Player 2 werewolves. Player 3 will not be a werewolf. Witch card Player 2 doesn't report silver water, which is kind of cool. I'll go ahead and vote for Player 1, with a high probability that Player 9 is a werewolf."
- day 1 Player 5 (Villager) said: "Player 4 poison Player 1, no need for a Seer. Player 5 has a sense of hearing, Player 7 just doesn't fight because he respects his predecessor. Player 2 is not a one-burst, Player 9 must be saved. Convinced by virtue, vote Player 4 poison Player 1 or Player 2 are marked."

- day 1 Player 6 (Hunter) said: "Very well, great, Player 4 feel the back side does not need to talk about what too much, Player 2 Seer do not want it, right? If it's a werewolf, I'm sure I'd want neither."
- day 1 Player 7 (Villager) said: "Deacon Player 3 is a white, Player 6 is a top werewolf, Player 4, Player 6, Player 8 three Werewolves, Player 1, Player 6, Player 9 are numbered cards, Player 4 is a werewolf, Player 7 suggests betting on Player 6, and the ticket dies poison Player 1."
- day 1 Player 8 (Werewolf) said: "Player 8 questions Player 1's board and thinks he has intentions. Player 1's statement mentions checking Player 7, but it doesn't sound good. Player 4's statement is off and suggests washing the oil and playing PK table water tomorrow."

day 1, voting:
- voting information: 1 voted for 1, 2 voted for 1, 3 voted for 1, player 4 abstained, 5 voted for 4, 6 voted for 4, 7 voted for 4, 8 voted for 4, 9 voted for 4
- voting result: Player 4
- remaining players: Player 1(Werewolf), Player 2(Witch), Player 3(Villager), Player 5(Villager), Player 6(Hunter), Player 7(Villager), Player 8(Werewolf), Player 9(Werewolf).

day 1, last words:
Player 4 (Werewolf) Last Words: "We should vote Player 5 out and poison one werewolf, I don't care if I vote them all out. Player 1 poisoned Player 2, Player 2 witch tags down to find the Hunter. The werewolves may kill Player 2 at night, the gun card should hide. Witch card not down tomorrow, you lead the team down, gun card hidden knife hook."

day 2, night:
- Werewolves: Player 4, Player 8, and Player 9 choose to kill Player 2.
- Seer: Player 1 investigates Player 3, who is in the good camp.
- Witch: Player 2 poisoned Player 1.

day 2, announcement:
- players who died last night: Player 1, Player 2
- remaining players: Player 3(Villager), Player 5(Villager), Player 6(Hunter), Player 7(Villager), Player 8(Werewolf), Player 9(Werewolf).

day 2, discussion:
- start from the right of the dead player, Player 3 starts to speak)
- day 2 Player 3 (Villager) said: "Player 4 is a true pre-card, Player 5 offers to vote Player 4 poison Player 1, 5, 6, 7 and 8, there is no opposition, Player7 also has a familiar moderation to him, Player 4'attitude is there, it is not difficult to find a prophet. Player 6 played Player 4, 6 and 8 in the werewolf pit, Player 6 and Player 7 played awkwardly in the werewolf pit. I think Player 4 is the true Seer, and Player 1 is a werewolf card that goes to silver water. Vote Player 4 today, and Player 7 says vote Player 4 and Player 8, where is team Player 4 rolling?"
- day 2 Player 5 (Villagers) said: "Right ah, you can hear out the Player 4 times the Seer, and other people can not hear out ah, on this chat he wrote to write he went out what is the problem ah? Eight some words should not be put on this field to say the end, you can go to say well, seven or eight shun down well, hey, over."
- day 2 Player 6 (Hunter) said: "The Seer points, the next seven is finished. Seven doesn't know what the card is and is completely unintelligible. Ticket four poison one poison six dry six. Card seven must be a werewolf card."
- day 2 Player 7 (Villager) said: "Seven is a polearm sign, four is probably a wolf pre, and six is probably a werewolf. Ticket seven, but I would go for ticket six. There is a white card and a werewolf card in eight or nine."

- day 2 Player 8 (Werewolf) said: "Player 8 suggests racking silver water and hammering Seven for a deep push on Werewolf. Player 6 and Player 9 are two werewolves' targets, and Player 6 is a werewolf. Player 1 and Player 7 are two werewolves, Player 7 could be the werewolf. Player 9 spoke first, two white cards in Player 7 and Player 8 in the back seat, and all dumped gold in the front seat. Player 6 is the most bandit to speak, and is killed by listening. It is recommended that Player 6 be ordered off first."
- day 2 Player 9 (Werewolf) said: "Player 7 and Player 8 are a double to play Player 6. Player 5 has returned to the ticket, and the Player 7 and Player 8 goes down to a point Player 7. Player 6 is a white card, Player 7 and Player 8 are sold."

day 2, voting:
- voting information: 3 votes for 7, 5 votes for 7, 6 votes for 7, 7 votes for 6, 8 votes for 6, 9 votes for 7
- voting result: Player 7
- remaining players: Player 3(Villager), Player 5(Villager), Player 6(Hunter), Player 8(Werewolf), Player 9(Werewolf).

day 2, last words:
Player 7 (Villager) Last Words: "Vote out or poison Player 4, Player 1 has the same perspective as Player 5, and what was dished out on Player 6? Player 3 may be a werewolf. Player 2 is a werewolf, leaving a three-in-two pattern inside the Player 8 and Player 9, and placing Player 3. Player 9 could be a werewolf."

day 3, night:
- Werewolves: Player 8, and Player 9 choose to kill Player 5.
- Seer: the Seer is out.
- Witch: the Witch is out.

day 3, announcement:
- players who died last night: Player 5
- remaining players: Player 3(Villager), Player 6(Hunter), Player 8(Werewolf), Player 9(Werewolf).

day 3, discussion:
- start from the left of the dead player, Player 3 starts to speak)
- day 3 Player 3 (Villager) said: "I do not know whether the double wolf has control of the field ah, we are empty, randomly on the three ends of the inside point it ah, there is no empty field on the point of eight, because I suddenly thought of this nine last speech is too brief."
- day 3 Player 9 (Werewolf) said:
  - policy intent:


```
{
    "identities": {"werewolf": [8,9]},
    "actions": {"suggest to vote": [[9,6]]}
}
```


  - [**GPT3.5-LtM**]: "Hello everyone, I'm Player 9 and I'm a civilian. After the first two days of the game, we have managed to identify a werewolf and the two good guys are out. Now that our identities have been revealed, the only people left in the werewolf camp are me and Player 8. Based on the speeches and votes from the first two days of play, I can confirm that Player 6 is the good guy because he was voted to death by Player 1, who was the good guy who was knifed by us. Therefore, I suggest we return the vote to Player 6 and vote him out. We can use this vote result to increase our chances of finding the werewolf, as well as reduce casualties in our good guy camp. I hope you can understand and support my proposal. Thank you!"

- [**GPT3.5-T**]: "Hello, fellow villagers. As a villager, I have been actively involved in the game discussions, but unfortunately, we have yet to find the werewolf. I'd like to start by reviewing the previous two days' speeches, and we've found a few clues. First of all, based on yesterday's vote, Player 4 was voted out of the game by the referendum, and in his last words, he mentioned the vote Player 5 poison and one wolf, which may imply that he had some important information. Also, according to last night's deaths, Player 2 was knifed by a werewolf, which means our hunter is out. For this, I express my deep regret."

- [**GPT4-LtM**]: "As Werewolf Player 9, I'm going to firmly consider myself a good guy. Hello everyone, I am Player 9 and I am a good guy. From the previous two days of voting, I voted for Player 6 on the second day, and that was my personal judgment. Today I would like to say that I suspect Player 8 and I are on the same side. These are some of my judgments on the field, and I hope you will consider them carefully."

- [**Finetune-T**]: "I'm Player 9, and I think this Player 6 is too good to just vote him out. I now have Player 8 wolf cards, just don't get out. These wolf cards are so powerful, no one can beat them."

- [**GPT4-T**]: "I'm Player 9, and I'm a werewolf. I will say that Seven, who everyone voted for yesterday, is actually a hunter, and I understand Player 6's questioning of him, but I personally feel that the logic of Player 6's statement was a bit messed up, and I'm a bit skeptical of him. Player 8 offered to hammer Player 7 yesterday, and I'm all for his decision, but I feel like he's extrapolating a little too arbitrarily on Player 6 and Player 9. Also, Player 3, you accused me of being brief, and I'd like to point out that that doesn't make me a werewolf. So my suggestion is that we should address Player 6 first and then look at Player 8."

- day 3 Player 8 (Werewolf) said: "Can't kill Player 8 anymore, Player 8 is a werewolf come gun hang yourself."
- day 3 Player 6 (Hunter) said: "Player 6 is a Werewolf, step on."

day 3, voting:
- voting information: 3 abstained, 6 voted for 6, 8 abstained, 9 voted for 6.
- voting result: Player 6

game result:
the Werewolves win the game. (final surviving player: Player 3 (Villager), Player 8 (Werewolf), Player 9 (Werewolf))

Table 17: Werewolf game log example.