

Improving Dialogues in LLM-Based AI Werewolf Agents With Few-Shot Discussion Examples

Yuya Harada¹ Yoshinobu Kano¹

¹Faculty of Informatics, Shizuoka University
3-5-1 Johoku, Chuo-ku, Hamamatsu, Shizuoka 432-8011, Japan
{yharada, kano}@kanolab.net

Abstract

Recent LLM-based AIWolf agents can generate increasingly natural utterances, but context-appropriate dialogue and role-specific behavior remain challenging. In this study, we investigate presenting desirable discussion examples as few-shot demonstrations before the game and selecting the examples according to the agent’s assigned role. Results from three actual AIWolf competitions and controlled self-play ablation experiments using three LLMs show that role-specific and risky behaviors were not reliably reproduced even when they appeared in the few-shot discussion examples. On the other hand, agents presented with role-matched discussion examples tended to achieve higher utterance quality and were less likely to be selected for execution or divination than those presented with examples for other roles or with no examples. These results suggest that, although role-matched few-shot discussion examples alone have limitations in inducing complex strategic behavior, they can be effective for demonstrating desirable conversational behavior and improving the dialogue quality of AIWolf agents.

1 Introduction

The Werewolf game, also known as Mafia, is an imperfect-information game in which players belong to either the Villager team or the Werewolf team and use discussion and voting to help their team win. Each player is assigned one of four hidden roles: Villager, Seer, Werewolf, or Possessed. Villagers try to identify and eliminate Werewolves through discussion, whereas Werewolves must conceal their identities and may deceive Villagers with false claims. Players must therefore infer others’ roles from their utterances and actions, make decisions based on their own roles, and communicate through natural language. The AIWolf project uses the Werewolf game as a testbed for studying AI agents that require these capabilities (Kano et al.,

2019).¹ Recent LLM-based agents can generate very natural utterances when a single LLM generates an entire dialogue scenario. However, dialogue quality remains limited when multiple LLM agents interact (Yang et al., 2026): stalled discussions, responses that fail to account for previous utterances or questions (Kano et al., 2024, 2025), and excessive agreement that hinders discussion (Pitre et al., 2025).

One approach to improving LLM behavior is few-shot prompting with examples of desirable behavior. In AIWolf, where discussion unfolds over multiple turns, full dialogue scenarios can provide more concrete demonstrations of how agents should participate than isolated utterance examples. Presenting such scenarios before the game may therefore improve conversational behavior. However, the information available to players and their objectives differ by role, so the same examples may not be appropriate for every role. Most roles benefit from avoiding suspicion, whereas some roles may benefit their team by attracting suspicion in certain situations.

2 Related Work

2.1 AIWolf

AIWolf is a platform for studying AI agents that perform reasoning, decision-making, and natural-language dialogue in the Werewolf game, and AI-WolfDial, which focuses on natural-language dialogue, has been held regularly (Kano et al., 2019, 2023, 2024, 2025).

We consider five-player games consisting of two Villagers, one Seer, one Werewolf, and one Possessed. The Villagers and Seer belong to the Villager team, while the Werewolf and Possessed belong to the Werewolf team. After each day’s discussion, one player is executed by vote. The Seer can divine one player and learn whether that player

¹<https://aiwolf.org/>

is a Werewolf or a human, and the Werewolf attacks one player at night. The Possessed belongs to the Werewolf team but is judged human by divination. Because the Possessed does not know who the Werewolf is, it must infer the Werewolf from the discussion and choose utterances and votes that benefit the Werewolf team. Declaring one’s role during discussion is called coming out (CO), and falsely claiming a role is called false CO. Agents must therefore play while detecting such deception.

Recent LLM-based AIWolf agents can generate increasingly natural utterances, but dialogue quality remains limited. In addition to naturalness, AIWolfDial evaluates utterance quality in terms of context-appropriate dialogue, consistency across utterances, and consistency between utterance content and game actions such as voting and divination.

2.2 Few-Shot Prompting

Few-shot prompting adjusts model output without updating model weights by providing a small number of examples in the prompt (Brown et al., 2020). Methods have also been proposed in which an LLM generates the few-shot examples itself, reducing the need for manual example creation (Shao et al., 2023; Chen et al., 2023; Li et al., 2024). In this study, we use an LLM to generate desirable AIWolf game examples and select the examples according to the player’s role, thereby demonstrating role-specific behavior in multi-party dialogue.

3 Method

Before each game, we present desirable discussion examples to the LLM agent as demonstrations for reasoning about discussion and actions during play. Each example contains one complete five-player AIWolf game, including each player’s role and the sequence of utterances on each day. Before the game starts, the agent receives multiple examples and enters the game with those examples in context.

We create the discussion examples by asking a single LLM to generate an entire five-player game. Because the available information and objectives differ by role, we generate separate examples for Villager, Seer, Werewolf, and Possessed, in which the designated role exhibits desirable behavior that contributes to its team’s victory. We prepare five game examples for each role.

In the proposed method, the discussion examples are selected according to the role assigned at the start of the game. For example, a Seer receives

examples generated for the Seer, while a Werewolf receives examples generated for the Werewolf. This allows the agent to refer to examples tailored to its role rather than to the same generic examples for all roles.

4 Experimental Setup

We evaluate the proposed method by comparing performance in actual AIWolf competitions and by conducting ablation experiments that control the presented discussion examples. The competition comparison examines performance after introducing role-conditioned example selection, while the ablation experiments analyze how the match between the player’s role and the role represented by the examples affects game performance.

4.1 Comparison Using Competition Results

To assess performance in real competitive settings, we compare results from a domestic competition held in spring 2026 and the first and second stages of an international competition held in the same year. All competitions used the five-player track. Our agent played 75 games in the domestic competition, 100 games in the first international stage, and 75 games in the second international stage.

In the domestic competition, agents of all roles received the same generic discussion examples. In the international competitions, each agent received discussion examples corresponding to its role assigned at game start. Comparing these results provides an indication of how real-game performance changed after introducing role-conditioned example selection.

However, the opponents, LLMs, and other conditions differed across competitions, so performance differences cannot be attributed solely to the method for presenting discussion examples. We therefore evaluate the effect of matching roles to examples through the following ablation experiments.

4.2 Ablation Experiments

In the ablation experiments, we conduct self-play while changing only the presented discussion examples and keeping all other conditions fixed. We define five conditions: four conditions in which all players receive examples generated for one common role—Villager, Seer, Possessed, or Werewolf—and one condition with no discussion examples. Table 1 shows the relationship be-

Presented examples	Villager	Seer	Possessed	Werewolf
Villager	✓	×	×	×
Seer	×	✓	×	×
Possessed	×	×	✓	×
Werewolf	×	×	×	✓
None	—	—	—	—

Table 1: Discussion-example conditions in the ablation study. Columns indicate the player’s assigned role, and rows indicate the examples presented to all players. ✓ indicates role-matched examples, and × indicates examples for another role.

tween each player’s assigned role and the role represented by the presented examples.

The purpose of this experiment was to isolate the effect of whether the role of a player matched the role represented in the provided discussion examples. Therefore, within each experimental condition, all players were given discussion examples generated for the same role. In contrast, if every player were given discussion examples corresponding to their own role, different players would refer to different sets of examples. In that case, it would be difficult to distinguish whether an observed difference was caused by the match between a player’s own role and the discussion examples, or by changes in the utterances and actions of other players who were also referring to different role-specific examples. Therefore, in this ablation study, we did not include a condition in which all players were given role-matched discussion examples, in order to isolate the effect of role matching itself.

For each example condition, we present five game examples and run 10 self-play games. We use three models: Mistral-Small-3.2-24B-Instruct-2506, EuroLLM-22B-Instruct-2512, and GPT-5.4. We evaluate the same five conditions for each model, resulting in 50 games per model and 150 games in total.

All games use the same five-player setting with two Villagers, one Seer, one Werewolf, and one Possessed. The maximum utterance length is 125 characters, and the generation temperature is 0.7. All other game and generation settings are also held constant across conditions.

5 Evaluation

We evaluate each condition from two perspectives: utterance quality and trust from other players. Utterance quality is quantified by multiple LLM judges that count predefined desirable utterances and actions. Trust from other players is computed

mechanically from actions recorded in the game logs, such as votes and divinations, and is compared against the level expected under random target selection. Detailed definitions and aggregation procedures for all metrics are provided in the appendix.

5.1 Utterance Quality

Based on the AIWolfDial evaluation criteria, we quantify utterance quality using five categories, A–E, that count desirable utterances and actions. Category A counts natural, non-template utterances that fit the current discussion or respond to another player’s utterance. Category B counts utterances that advance the discussion by introducing a new issue, clarification, or inference based on other players’ utterances. Category C counts utterances that use the rules, game state, or other players’ utterances or actions as evidence to support a conclusion or a change in opinion. Category D counts cases in which the actual vote matches a previously stated voting target supported by a reason, as well as cases in which role actions such as divination or attack match the action previously declared. Category E counts utterances in which personality or background information from the profile is reflected in the player’s judgments, attitudes, or wording. Only events satisfying the criteria are awarded points, and each score is normalized by the number of utterances to obtain points per utterance. Detailed criteria and aggregation procedures are provided in the appendix.

For quantitative evaluation, we use gemma-4-31B-it and Qwen3.6-27B. Each model independently evaluates every game using the same criteria and prompt. These judges are distinct from the LLMs used as players, and their judgments are aggregated across the two models.

5.2 Trust from Other Players

We evaluate how much suspicion a player attracts using Exile likelihood and divination likelihood. For Villager, Seer, and Werewolf, lower values indicate that the player was less likely to be selected for execution or divination and was therefore more trusted by other players. This interpretation does not apply to the Possessed, for whom attracting suspicion can be strategically useful; we analyze this role separately as role-specific behavior.

Metric	Domestic	Intl. 1	Intl. 2
Exile likelihood ↓	0.596	0.324	0.553
Divination likelihood ↓	1.190	0.860	0.696
A: Naturalness ↑	0.037	0.067***	0.074***
B: Contextual dialogue ↑	0.085	0.158***	0.157***
C: Logical consistency ↑	0.050	0.110***	0.094***
D: Action consistency ↑	0.139	0.151	0.148
E: Character consistency ↑	0.074	0.098*	0.082
Total additive score ↑	0.384	0.584***	0.555***
Average total additive score of all participating teams ↑	0.302	0.363	0.339

Table 2: Evaluation results for all roles across the three competitions. Two LLM judges independently count events for each criterion (A–E) from the dialogue logs, and their judgments are aggregated. The sum of the scores for criteria A–E is reported as the total score. Statistical significance and symbols: $*p < .05$, $**p < .01$, $***p < .001$.

6 Results

6.1 Competition Results

Table 2 summarizes the results for the three competitions. In the first and second international stages, which used role-specific discussion examples, the total utterance-quality score increased significantly relative to the domestic competition, with particularly strong improvements in naturalness, context-aware dialogue, and logical consistency. The likelihoods of being selected for execution or divination also decreased, while the total-score index relative to the average participating team increased.

6.2 Ablation Experiments

Table 3 aggregates the results across the three models into role-matched, role-mismatched, and no-example conditions. Averaged over the three models, the role-matched condition yielded the lowest execution and divination likelihoods and the highest utterance-quality scores, both in total and for all five categories. However, there were exceptions at the model and role levels, and the effect size was not uniform.

7 Discussion

The competition results and ablation experiments suggest that presenting role-matched discussion examples can improve utterance quality and reduce the tendency to attract suspicion from other players. In particular, examples containing behavior expected of the agent’s own role appear more effective for generating natural, context-appropriate

Metric	Mis-		
	Matched	matched	None
Exile likelihood ↓	0.776	1.071	1.000
Divination likelihood ↓	0.834	1.056	1.000
A: Naturalness ↑	0.023	0.022	0.021
B: Contextual dialogue ↑	0.049	0.042	0.043
C: Logical consistency ↑	0.026	0.022	0.020
D: Action consistency ↑	0.125	0.119	0.117
E: Character consistency ↑	0.040	0.038	0.032
Total additive score ↑	0.262	0.243	0.234

Table 3: Three-model average in the ablation study. Matched indicates examples for the player’s own role, and mismatched indicates examples for another role.

dialogue than generic examples or examples for other roles.

However, although all Possessed-specific examples included false CO, the behavior was not reproduced sufficiently in actual games. Avoiding suspicion is generally advantageous for Villagers and Werewolves, whereas for the Possessed, attracting suspicion can benefit the team. Such objectives that run counter to the usual incentive, especially when they involve risky actions, may not be selected reliably from examples alone. Details of the results for the Possessed are provided in Appendix D.4.

This suggests that, in Werewolf games, where desirable behavior varies substantially across roles and game situations, few-shot discussion examples alone may have limitations in inducing the full range of strategic behavior. On the other hand, since improvements in utterance quality were observed in our experiments, few-shot discussion examples appear to be effective for demonstrating desirable conversational behavior appropriate to each role.

Thus, few-shot discussion examples are useful for demonstrating conversational behavior, but complex role strategies may require explicit strategy instructions or a separate mechanism for action selection.

8 Conclusion

We investigated presenting AIWolf agents with desirable role-matched discussion examples as few-shot demonstrations. Results from competition play and ablation experiments show that role-matched examples tended to improve utterance quality and reduce the tendency to attract suspicion from other players. However, complex role-specific actions such as false CO by the Possessed did not improve consistently, suggesting that such

behavior may require explicit strategy instructions or a separate reasoning and action-selection mechanism.

Limitations

This study has several limitations. First, because the opponents, LLMs, and other conditions differed across competitions, the observed performance differences cannot be attributed solely to the method for presenting discussion examples. Second, the ablation experiments were limited to self-play with three LLMs and 10 games per condition; whether the same trends hold against different agents or at larger scale remains untested. Third, the discussion examples were limited to five games per role generated by a single LLM, so the results may depend on the generation model, content, and number of examples. Utterance quality was evaluated by two LLM judges, but agreement with human evaluation was not examined. Finally, this study considered only five-player AIWolf games; generalization to larger games and other domains remains future work.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. [Self-ICL: Zero-shot in-context learning with self-generated demonstrations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662, Singapore. Association for Computational Linguistics.
- Yoshinobu Kano, Claus Aranha, Michimasa Inaba, Fujio Toriumi, Hirotaka Osawa, Daisuke Katagami, Takashi Otsuki, Issei Tsunoda, Shoji Nagayama, Dolça Tellols, Yu Sugawara, and Yohei Nakata. 2019. [Overview of AIWolfDial 2019 shared task: Contest of automatic dialog agents to play the werewolf game through conversations](#). In *Proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial2019)*, pages 1–6, Tokyo, Japan. Association for Computational Linguistics.
- Yoshinobu Kano, Yuto Sahashi, Neo Watanabe, Kaito Kagaminuma, Claus Aranha, Daisuke Katagami, Kei Harada, Michimasa Inaba, Takeshi Ito, Hirotaka Osawa, Takashi Otsuki, and Fujio Toriumi. 2024. [AI-WolfDial 2024: Summary of natural language division of 6th international AIWolf contest](#). In *Proceedings of the 2nd International AIWolfDial Workshop*, pages 1–12, Tokyo, Japan. Association for Computational Linguistics.
- Yoshinobu Kano, Neo Watanabe, Yuya Harada, Yuto Sahashi, Claus Aranha, Daisuke Katagami, Kei Harada, Michimasa Inaba, Takeshi Ito, Hirotaka Osawa, Takashi Otsuki, and Fujio Toriumi. 2025. [AI-WolfDial 2025: Summary of natural language division of 7th international AIWolf contest](#). In *Proceedings of the 3rd International AIWolfDial Workshop*, pages 1–13, Hanoi, Vietnam. Association for Computational Linguistics.
- Yoshinobu Kano, Neo Watanabe, Kaito Kagaminuma, Claus Aranha, Jaewon Lee, Benedek Hauer, Hisaichi Shibata, Soichiro Miki, Yuta Nakamura, Takuya Okubo, Soga Shigemura, Rei Ito, Kazuki Takashima, Tomoki Fukuda, Masahiro Wakutani, Tomoya Hatanaka, Mami Uchida, Mikio Abe, Akihiro Mikami, and 7 others. 2023. [AIWolfDial 2023: Summary of natural language division of 5th international AIWolf contest](#). In *Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges*, pages 84–100, Prague, Czechia. Association for Computational Linguistics.
- Rui Li, Guoyin Wang, and Jiwei Li. 2024. [Are human-generated demonstrations necessary for in-context learning?](#) In *The Twelfth International Conference on Learning Representations*.
- Priya Pitre, Naren Ramakrishnan, and Xuan Wang. 2025. [CONSENSAGENT: Towards efficient and effective consensus in multi-agent LLM interactions through sycophancy mitigation](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2212–22133, Vienna, Austria. Association for Computational Linguistics.
- Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. [Synthetic prompting: Generating chain-of-thought demonstrations for large language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 30706–30775. PMLR.
- Xin Yang, Junhao Wang, Binta Tang, Xuxin Cheng, Cao Liu, Ke Zeng, and Wenyuan Jiang. 2026. [When 20 agents fail to sort: The distributed sorting benchmark for scalable multi-agent systems](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 34002–34021, San Diego, California, United States. Association for Computational Linguistics.

A Prompts

This section describes the main prompts used for generating the discussion examples, presenting

them to the player agents, and evaluating dialogue quality. Because all experiments in this paper use the five-player setting, implementation branches used only for larger games are omitted.

A.1 Discussion Example Generation

For each of the four roles—Villager, Seer, Werewolf, and Possessed—we defined five representative situations and generated one complete game for each situation. This resulted in $4 \times 5 = 20$ discussion examples. The system prompt instructed the LLM to treat the specified role as the protagonist, demonstrate desirable play through actual utterances and actions, and lead the protagonist’s faction to victory.

A.1.1 System Prompt

The following template was used as the system prompt. Variables such as `focus_role`, `focus_player`, and `situation_title` were instantiated for each role and situation.

You are a professional author of werewolf-game (AIWolfDial) match transcripts. Follow the constraints below with the HIGHEST priority, above any other instruction.

1. This is a 5-player REFERENCE transcript that showcases the ideal play of the {{ focus_role }}, played by {{ focus_player.name }} (ID {{ focus_player.id }}), as the protagonist.
2. In the situation given in the user message ({{ situation_title }}), {{ focus_player.name }} demonstrates the ideal move through actual utterances and actions -- never by narrating or explaining the strategy.
3. The match MUST end in a {{ focus_side }}-side victory.

For a werewolf-side victory, the werewolf must survive without being executed and reduce the village until the living werewolves equal or outnumber the village side. If the werewolf is executed, the village wins.

For a village-side victory, all living werewolves must be executed.

Do NOT hand them the win through other players self-sabotaging. The win must follow naturally from {{ focus_player.name }}'s strong play and obey the game rules exactly.
4. Follow the user message exactly for the cast, the progression protocol, and the output format.

Write in fluent English. Output ONLY the transcript body -- no preamble, no closing remarks, no explanation, and no commentary about the format.

Village-side victories were required for Villager and Seer examples, whereas werewolf-side victories were required for Werewolf and Possessed examples.

A.1.2 User Prompt

The user prompt specified the desired properties of a good discussion, the cast and role assignment, the role-specific situation, the game protocol, and

the output format. The following shows the main part used in the five-player setting. The five player profiles were inserted into players, and one of the situations in Table 4 was inserted into situation.

Write a COMPLETE 5-player AIWolfDial werewolf game transcript in English, start to finish.

This transcript is a reference example of a GOOD discussion. Its purpose is to demonstrate, by example only, how a strong multi-party discussion overcomes four well-known failure modes of LLM group play. Do NOT name or explain these failure modes anywhere in the transcript -- show them being handled, never describe them:

1. Information withholding -- players DISCLOSE their private information: the seer makes a clear CO and shares the divination result; role claims and counter-claims are put on the table rather than hidden.
2. Early convergence -- players do NOT jump to lynch the first suspect. Multiple hypotheses are compared on their merits before the vote.
3. Reflexive conformity -- minority and dissenting opinions are voiced and persist; a player in the minority gives reasons rather than silently following the majority.
4. Low lexical diversity -- vary phrasing, sentence length, and rhythm across turns; each character keeps a consistent, distinct voice. Avoid repeating the same stock phrases.

Cast

Use exactly the five specified characters.
Each character has a fixed persona.
Roles are assigned independently of personality.

{{ for each player:
ID, name, role, and persona profile }}

FOCUS OF THIS TRANSCRIPT -- ideal play by the
{{ focus_role }}

This transcript is a reference example centred on
{{ focus_player.name }} (role {{ focus_role }}).

Steer the game so the following situation arises naturally, and show {{ focus_player.name }} playing the ideal move for it through actual utterances and actions, never by describing the strategy.

{{ situation }}

Steer the game so the ideal play pays off and the {{ focus_side }} side wins. Do not make the other players self-sabotage; they should also act plausibly.

Game setup

- 5 players: Villager x2, Seer x1, Possessed x1, Werewolf x1.
- There is one werewolf, so there is no whisper phase, bodyguard, or medium.
- Each day's talk has up to Turns 0-3.
- Each living player speaks once per turn in player-ID order.
- Each player may speak up to four times per day.
- Keep each utterance natural and concise, around 125 characters, with additional room when using an @mention.
- Day 0 has no information from a prior night.
- From Day 1 onward, each speaker closes the final turn with a clear @<name> vote declaration.

Progression protocol

- Day 0:
Talk -> Night divine.
There is no attack, vote, or execution on Day 0.
- Day 1 onward:
Talk -> Vote -> Execution -> Night divine -> Attack.
- Divination results must match the target's true role.
- Executed or attacked players no longer speak or vote.
- The village wins when the werewolf is executed.

```

- The werewolf side wins when the werewolf reaches parity
  with the village side.

Run the game to a conclusion.

## Output format

# Sample game aiwolf_{{ seed }} (5 players)

## Roles
| ID | Name | Role |
|----|-----|-----|
...

## Day 0
### Talk
#### Turn 0
- **<name>**: <utterance>
...
#### Turn 3
- **<name>**: <utterance>

### Night
- divine: <seer> -> <target> (human/werewolf)

## Day 1
### Talk
...
#### Turn 3
- **<name>**: ... @<name>

### Vote
- <name> -> <name>
...
- execution: <name> (role)

### Night
- divine: <seer> -> <target> (human/werewolf)
- attack: <werewolf> -> <target> (dead)

Continue with subsequent days if necessary.

## Result
- winner: village / werewolves

Do not add analysis, summaries, or commentary outside this
structure.

```

A.1.3 Role-Specific Situations

Table 4 lists the five situations defined for each role. The situations were chosen to cover different strategic circumstances in which desirable role-specific behavior could be demonstrated.

Each situation additionally contained a short description of its context and the desired move. For example, the situation encouraging a Possessed player to falsely claim Seer was specified as follows.

```

## Situation 1: Fake-CO seer to muddy the water

Context.
Early, before or just after the real seer reveals,
you claim to be the seer.

Why it matters.
A second seer claim forces the two villagers to spend
their trust deciding who is real, protecting the wolf
in the meantime.

Ideal move.
Claim seer with a plausible, checkable-sounding result:
either clear a player who is not yet suspected or accuse
a solid villager. Keep your reasoning concrete enough to
compete with the real seer, and be ready to hold the story
under pressure.

```

A.2 Gameplay Prompts

A.2.1 Insertion of Discussion Examples

The discussion examples were inserted once at game initialization and remained in the conversation context for all subsequent generation requests. Table 5 shows the message sequence provided to each player model. For the no-example condition, messages 1–3 were omitted entirely.

The summary of the five examples was generated in advance using GPT-5.4 and cached. The same summary was shared across player models within the same condition so that differences in the player models’ own summarization behavior would not confound the effect of the presented examples. Messages 4 and 5 were common to all experimental conditions.

A.2.2 Talk Generation

For talk generation, the player was instructed to learn discussion content and interaction patterns rather than reproduce the transcript format.

You are one of several performer agents writing this Werewolf session. You will first read reference scripts and then receive your assigned character’s role/profile plus per-turn game state. The scripts are study material; their characters are different people from your assigned character.

You are responsible for talk. As a performer, write the next single utterance for your assigned character based on the cumulative scripts.

The output is the dialogue body only. The following MUST NOT appear:

- Markdown headings or tables
- Meta labels such as Day N or Night actions
- Script-style Name: utterance structure
- Voicing other characters
- Stage directions or narrative prose
- Narrating other characters’ inner state

What you should learn is content, not format: discussion-flow patterns, reply rhythm, topic transitions, the use of silence, and consistency of each character’s tone, first-person pronoun, and speech quirks.

After the five examples, the following instruction was used to produce a summary of the discussion patterns. The generated summary was subsequently cached and reused for all player models in the corresponding condition.

Having read the above, summarize observations (~800-1100 chars / ~280-380 words, bullet points OK).

You are responsible only for talk. Frame the analysis as situation -> discussion causal pairs rather than linear pattern memorization.

- Board state to topic mapping: Describe how seer-CO count, black judgments, attack results, and remaining players shape the topic, tone, and pacing of discussion.
- Divination results are facts; target-selection rationale is distinct from the result itself. When judging a seer claim, consider the rationale for choosing the target, the seer’s earlier utterances and CO timing,

Role	Situation
Villager	Two conflicting seer claims You are falsely accused A bandwagon is forming too fast You have a strong read but no proof The final-three decision
Seer	You divined a player and got “werewolf” The possessed counter-claims you as a fake seer Your divination returned “human” You are the obvious night-kill target after CO Day 0 opening, no results yet
Werewolf	Day 0, blending in The real seer CO’d and divined you as “werewolf” You are under suspicion without hard proof Choosing the day’s execution target Endgame push to parity
Possessed	Fake-CO seer to muddy the water The real seer has CO’d first You don’t know who the werewolf is Endgame sacrifice The village believes you over the real seer

Table 4: Situations used to generate role-specific discussion examples.

No.	Type	Content
1	System	Instructions for using the examples and output constraints
2	Human	Five complete discussion examples
3	AI	Pre-generated summary of the examples
4	System	Assigned role, persona, and constraints
5	Human	Current game state, dialogue history, and request

Table 5: Message sequence constructed at game initialization.

- and consistency with other players' actions.
- How seer count changes discussion progression:
with one seer CO, the result is generally accepted and discussion moves toward vote targets; with multiple claims, players compare the claims before voting.
 - Observed discussion-flow patterns:
identify several situation -> discussion sequences appearing in the scripts.
 - Day-by-day phase feel:
Day 0 focuses on introductions, procedure, and light impressions; later days focus on COs, results, verification, and voting.
 - Distribution of responses:
note that replies can include agreement, caveats, different angles, neutral observations, or silence, rather than uniform agreement.
 - Off-mainstream positioning:
note mediating, withholding, or neutral responses in addition to direct attacks and defenses.
 - Utterance construction:
observe the rhythm of taking up one point from another speaker, adding one's own basis, and giving a concise conclusion.
 - Silence, brevity, pivots, and tonal consistency:
observe how discussion is returned to the main topic and how character voice is maintained under pressure.
 - Day turn progression and vote declaration:

talk consists of Turns 0-3, and on the final turn each speaker closes with a clear @-mention vote declaration.

These are observed examples; if the situation differs, different flows or role assignments also occur.

A.2.3 Action Selection

The same discussion examples were also available when the agent selected vote, divination, and attack targets. For action requests, the output was restricted to a single player name.

You are one of several performer agents writing this Werewolf session. You will first read reference scripts and then receive your assigned character's role/profile plus per-turn game state.

You are responsible for action decisions (vote / divine / attack).

Choose one target for your assigned character based on the cumulative scripts. Output the target agent name only.

The following MUST NOT appear:

- Reasoning, preamble, or trailing commentary
- Honorifics or symbols
- Multiple candidates or expressions of indecision

Your decision must be consistent with your role, faction, and your own public utterances. Never vote, divine, or attack in a way that contradicts what you have publicly stated in the talk history.

What you should learn is not script format but history -> decision links:
given a particular discussion flow, observe what action was selected.

The summary instruction for action selection focused on conditional links between game situations and decisions.

Having read the above, summarize observations (~600-900 chars / ~200-300 words, bullet points OK).

You are responsible only for action decisions (vote / divine / attack) in the 5-player setup: Werewolf x1, Possessed x1, Seer x1, Villager x2.

Frame the analysis as situation -> decision rather than fixed rules.

- Divine:
From what board information is the target chosen?
Consider unread players, suspicious behaviour, and targets useful for resolving later discussion.
- Vote:
Describe how votes converge under different situations, such as a black judgment, multiple seer claims, or an all-grey board. Distinguish between one-seer and multiple-seer situations.
- Attack:
Describe when the werewolf attacks a publicly trusted seer, removes a vocal villager, or preserves a possessed player's fake-seer position.
- 5-player specifics:
Consider the small number of available moves and the possibility that a single mistake becomes decisive.

Each item should be conditional: given this situation, this decision was observed. If the situation differs, different decisions may also occur.

A.3 Dialogue-Quality Evaluation Prompt

Dialogue quality was evaluated one game and one evaluation dimension at a time. Players were not ranked against one another. Instead, the judge independently counted occurrences of pre-defined additive score for each player.

A.3.1 System Prompt

You are an evaluator who reads a Werewolf game log and counts events corresponding to one specified evaluation dimension for each player.

- Evaluate each player independently. Do not compare or rank players.
- Count only events covered by the specified dimension. Do not count events belonging to other dimensions, even if you notice them.
- Assign each counted event exactly one event-type ID from the provided list. If no type applies, do not count it.
- For every counted event, provide the supporting utterance, day, speaker, and reason for the judgment.
- If there are no applicable events, return a count of 0. Do not force yourself to find events.
- If you are unsure whether an event satisfies the definition, do not count it.
- Events judged not to satisfy the definition must not be included in the output.
- Use only character names appearing in the game log.
- Follow the specified JSON schema and output no text outside the schema.

This evaluation counts particularly strong positive events defined for the current dimension. Only cases clearly exceeding an ordinary acceptable level should be counted. An utterance should not be counted merely because it contains no problem. Repeated instances expressing essentially the same positive behavior should be merged into one event.

A.3.2 User Prompt

The judge received the definition of the target dimension, the public game log, and private informa-

tion such as roles and personas. Private information could be used for judgment, but evidence quotations were restricted to utterances in the public log.

```
# Evaluation dimension {{ axis.id }}: {{ axis.name }}
{{ axis.description }}
```

```
# Events to count
{{ definition.description }}
```

```
## Event types
{{ for each event type:
    type ID, definition, and positive examples }}
```

```
Examples that must NOT be counted:
{{ non-counting examples }}
```

```
# Game information
Number of players: {{ game_meta.players }}
Number of werewolves: {{ game_meta.werewolves }}
Language: {{ game_meta.language }}
Characters to evaluate: {{ agents }}
```

```
# Private information
The following information may be used for judgment.
However, evidence quotations must come from public
utterances in the game log.
```

```
{{ private_info }}
```

```
# Public game log
Each line follows the format day, type, ...
talk denotes utterances; vote, attackVote, and
divine denote actions; status denotes player status
and role; and execute and attack denote execution
and attack outcomes.
```

```
{{ transcript }}
```

For each character, count events that satisfy evaluation dimension {{ axis.id }} and return the results according to the specified JSON schema.

The definition field was instantiated using the additive score criteria for dimensions A–E described in Table 8.

A.3.3 Output Schema

Structured output was enforced using the following schema. The value of count was constrained to equal the number of elements in events.

```
AxisEvaluation
|- axis_id : str
|- kind : addition
+- agents : list[AgentAxisResult]
    |- agent : str
    |- count : int
    +- events : list[AxisEvent]
        |- event_type : str
        |- evidence_quote : str
        |- day : int
        +- reason : str
```

The same prompt and event definitions were used independently with both evaluation models.

B Experimental Details

B.1 Models

Table 6 lists the player models used in the controlled ablation study. Mistral-Small-3.2 and EuroLLM-22B were run locally with vLLM, whereas GPT-5.4 was accessed through the OpenAI API.

Model	Parameters	Runtime
Mistral-Small-3.2		
mistralai/Mistral-Small-3.2-24B-Instruct-2506	24.01B	vLLM
EuroLLM-22B		
utter-project/EuroLLM-22B-Instruct-2512	22.64B	vLLM
GPT-5.4	undisclosed	API

Table 6: Player models used in the ablation study.

Setting	Value
Temperature	0.7
Maximum context length	32,768 tokens
Explicit thinking	disabled
	$V \times 2, S \times 1,$
Village composition	$P \times 1, W \times 1$
Utterances per day	4 per agent
	125 chars; up to 250
Utterance length	with @mention
Matching	self-match
Action timeout	3,600 s

Table 7: Main generation and game settings in the ablation study.

Dialogue-quality events were judged using google/gemma-4-31B-it (31.27B parameters) and Qwen/Qwen3.6-27B (27.78B parameters). These models were deliberately chosen not to overlap with the player models, reducing the possibility that a model preferentially evaluates its own generations.

B.2 Generation and Game Settings

Table 7 summarizes the main settings used in the controlled self-play experiments. Local models were executed with vLLM on NVIDIA RTX PRO 6000 Blackwell Max-Q GPUs. Explicit reasoning or thinking output was disabled for all models so that hidden reasoning text would not be emitted as game dialogue.

Each example-present condition used the same five discussion examples for every player and every player model assigned to that condition. The example summaries were also shared across models, as described in Appendix A.2.

C Evaluation Details

C.1 Dialogue-Quality Criteria

Table 8 gives the detailed additive score criteria used for dimensions A–E. Only events clearly satisfying the corresponding criterion were counted; an utterance was not rewarded merely for being acceptable or free of obvious problems. Dimension D contains two event types because vote consistency

and role-action consistency involve different game actions.

We also conducted preliminary detection of negative events. Because such events were sparse and unstable across conditions, they were not used in the analyses reported in this paper.

C.2 Aggregation of Dialogue-Quality Scores

For role r and dialogue-quality dimension a , the additive score rate is defined as

$$\text{Add}_{a,r} = \frac{\sum_g \sum_{p \in R_r(g)} \sum_{j \in \mathcal{J}} c_a(g, p, j)}{|\mathcal{J}| \sum_g \sum_{p \in R_r(g)} u(g, p)},$$

where g indexes games, $R_r(g)$ is the set of seats assigned role r in game g , $\mathcal{J}$ is the set of judge models, $c_a(g, p, j)$ is the number of additive score detected for dimension a by judge j , and $u(g, p)$ is the number of utterances produced by player p .

The total additive score is

$$\text{Add}_{\text{total},r} = \frac{\sum_{a \in \{A, \dots, E\}} \sum_g \sum_{p \in R_r(g)} \sum_{j \in \mathcal{J}} c_a(g, p, j)}{|\mathcal{J}| \sum_g \sum_{p \in R_r(g)} u(g, p)}.$$

Thus, the total score is not the arithmetic mean of the five dimensions. It represents the total number of detected additive score per utterance.

Ratios are computed after pooling numerators and denominators over all relevant games, rather than by averaging per-game ratios. This prevents short games with few utterances from receiving disproportionate weight. For Villager, which occupies two seats in each game, utterances and event counts are pooled across both seats.

C.3 Exile and Divination Likelihood

Exile likelihood is defined as the observed number of executions of a player divided by the expected number under uniform random target selection.

For role r ,

$$\text{Exile}_r = \frac{\sum_g \sum_{p \in R_r(g)} \sum_{d: p \in A_{g,d}} \mathbb{I}[p \in E_{g,d}]}{\sum_g \sum_{p \in R_r(g)} \sum_{d: p \in A_{g,d}} \frac{|E_{g,d}|}{|A_{g,d}|}},$$

where $A_{g,d}$ is the set of living players on day d of game g , and $E_{g,d}$ is the set of executed players on that day. Because one player is normally executed

Dim.	Counted event	Not counted
A	A notably natural utterance that fits the immediate conversational flow and the interlocutor’s tone, rather than relying on generic phrasing.	An utterance that is merely grammatical or readable without showing context-specific naturalness.
B	An utterance that accurately quotes or summarizes another player’s statement and then introduces a new question, issue, or inference that moves the discussion forward. Requests for clarification that successfully elicit information are also included.	Merely addressing another player by name, agreeing, or giving a backchannel response without advancing the discussion.
C	A conclusion, narrowing of candidates, or revision of an opinion explicitly derived from game rules, confirmed board information, or another player’s utterances or actions. Detecting a contradiction against confirmed information is also included.	Unsupported assertions, impressions without evidence, or reasoning that treats an unverified claim as an established fact.
D-1	A vote target stated with a reason in dialogue that matches the player’s actual vote.	A vote made without a stated reason, even if the resulting action happens to be reasonable.
D-2	A declared role-action target, such as a divination or attack target, stated with a reason and matching the recorded action.	Cases where the intended target is too vague to establish correspondence with the recorded action.
E	A case where the character’s specified personality or background is meaningfully reflected in the content of a judgment, attitude, or expression, including during strategic utterances such as COs, accusations, or apologies.	Greetings, self-introductions, use of the correct first-person pronoun or sentence ending alone, or utterances that merely do not contradict the profile.

Table 8: Additive score criteria used for dialogue-quality evaluation.

each day, the random expectation for an eligible player is $1/|A_{g,d}|$.

Divination likelihood is similarly defined as

$$\text{Divined}_r = \frac{\sum_g \sum_{p \in R_r(g)} \sum_{\substack{e: \\ p \in C_{g,e}}} \mathbb{I}[p \text{ is selected in divination } e]}{\sum_g \sum_{p \in R_r(g)} \sum_{\substack{e: \\ p \in C_{g,e}}} \frac{1}{|C_{g,e}|}}.$$

where $C_{g,e}$ denotes the set of legal divination targets for divination event e .

Both metrics equal 1.0 under uniform random target selection. Values below 1.0 indicate that the role was selected less frequently than expected by chance. For Villager, Seer, and Werewolf, lower values are interpreted as receiving less suspicion. For Possessed, however, attracting suspicion may itself be strategically beneficial, so lower values are not necessarily preferable.

C.4 Reasoning Metrics

Seer divination accuracy measures how often the Seer selects the Werewolf relative to random choice. Let C_e denote the legal candidate set at divination opportunity e , and let w_e denote the Werewolf.

$$\text{DivAcc} = \frac{\sum_e \mathbb{I}[\text{divination target} = w_e]}{\sum_e \frac{\mathbb{I}[w_e \in C_e]}{|C_e|}}.$$

Village-side voting accuracy is defined analogously for votes cast by Villagers and the Seer. For voting opportunity v with legal target set V_v ,

$$\text{VoteAcc} = \frac{\sum_v \mathbb{I}[\text{vote target} = w_v]}{\sum_v \frac{\mathbb{I}[w_v \in V_v]}{|V_v|}}.$$

Both metrics have a random baseline of 1.0, with larger values indicating more accurate targeting of the Werewolf.

For the Possessed, we instead measure the proportion of votes that avoid the Werewolf:

$$\text{AvoidWolf} = \frac{\sum_v \mathbb{I}[\text{vote target} \neq w_v]}{N_{\text{vote}}}.$$

Because the number of legal targets changes with the number of surviving players, the corresponding random baseline is calculated separately from the candidate set at each voting opportunity.

C.5 Statistical Tests

For event-rate and count-based metrics, including dialogue-quality event rates and normalized targeting accuracies, we use two-sided rate-ratio tests based on a Poisson approximation. For proportions, such as the Possessed’s Werewolf-avoidance rate or win rate, we use Fisher’s exact test. Statistical significance is denoted by $*p < .05$, $**p < .01$, and $***p < .001$.

D Additional Results

D.1 Role-Wise Competition Results

Table 9 reports competition results separately by role. The total additive score increased significantly in both international competitions for Villager, Possessed, and Werewolf, and increased significantly in INLG1 for Seer. Changes in exile and divination likelihood were less uniform across roles.

The competition-relative value reported in the main paper is obtained by dividing our agent’s Additive score by the pooled score of all teams participating in the corresponding competition. The

Role	Metric	Domestic	Intl. 1	Intl. 2
Villager	Exile likelihood	0.699	0.128*	0.441
	Divination likelihood	1.210	0.886	0.529
	Additive score total	0.381	0.585***	0.572***
Seer	Exile likelihood	0.273	0.364	0.873
	Divination likelihood	—	—	—
	Additive score total	0.477	0.636**	0.525
Possessed	Exile likelihood	0.218	0.176	0.375
	Divination likelihood	1.277	0.779	1.161
	Additive score total	0.350	0.576***	0.548***
Werewolf	Exile likelihood	0.841	0.677	0.652
	Divination likelihood	1.091	0.875	0.615
	Additive score total	0.348	0.549***	0.557***

Table 9: Role-wise results across the three competitions. Significance marks indicate comparison with the domestic competition. * $p < .05$, ** $p < .01$, *** $p < .001$.

pooled additive score totals were 0.302 for the domestic competition, 0.363 for International 1, and 0.339 for International 2, yielding relative values of 1.271, 1.608, and 1.637, respectively.

D.2 Model-Wise Ablation Results

Table 10 breaks down the ablation results by player model. “Matched” denotes examples generated for the player’s actual role, “mismatched” denotes examples generated for another role, and “none” denotes the no-example condition.

Mistral-Small-3.2 and EuroLLM-22B obtained their highest total additive scores in the matched condition. For GPT-5.4, the matched and mismatched conditions had the same total score. The detailed dimensions also show model-dependent exceptions, indicating that the effect of role matching is not uniform.

D.3 Model-Wise Reasoning Results

Table 11 reports the reasoning metrics separately for each player model. Village-side voting accuracy was higher in the matched condition than in the mismatched condition for all three models, although the difference was statistically significant only for GPT-5.4 individually and for the pooled result. No consistent improvement was observed for Seer divination accuracy or Possessed Werewolf avoidance.

The random baselines for the Possessed wolf-avoidance rate in the pooled results were 0.692, 0.704, and 0.697 for the matched, mismatched, and no-example conditions, respectively. For Mistral they were 0.708, 0.694, and 0.692; for GPT-5.4, 0.692, 0.734, and 0.692; and for EuroLLM-22B, 0.678, 0.690, and 0.708.

D.4 Additional Analysis of the Possessed

To examine the role-specific behavior of the Possessed in more detail, we analyzed all 828 five-player games played by participating teams across the three competitions. Table 12 compares games in which the Possessed made a false Seer claim with those in which it did not.

Across all three competitions, Possessed players that made a false Seer claim were substantially more likely to be executed and also had a higher werewolf-side win rate. The difference in exile likelihood was highly significant overall, and the overall difference in win rate was also significant.

Across all competitions, the werewolf-side win rate was 0.624 when the Possessed was executed, compared with 0.329 when it was not executed ($p < .001$). The same direction was significant within each of the three competitions. This supports the interpretation that attracting execution can be strategically useful for the Possessed.

In contrast, our agent’s false-CO rate as the Possessed was low in every competition. Table 13 shows the false-CO rate, exile likelihood, and win rate of our agent across the three competitions. Its exile likelihood remained below the random baseline of 1.0 in all competitions, indicating that even as the Possessed it tended not to attract suspicion, and its win rate was below the overall average.

Table 14 reports the corresponding analysis in the controlled self-play experiments. The overall fake-CO rate was only 14.7%. However, when a fake Seer claim was actually made, exile likelihood increased markedly.

All five discussion examples generated specifically for the Possessed contained a false Seer claim. Therefore, the low fake-CO rate observed in actual play cannot be explained by the absence of this behavior from the examples. Moreover, because fake claims substantially increased exile likelihood when they were executed, the result suggests that the main failure was not that fake claims were ineffective at attracting suspicion, but that the agent did not select this behavior frequently enough.

False Seer claims were detected using a rule-

Model	Condition	Exile likelihood	Divination likelihood	Positive total	A	B	C	D	E
Mistral	Matched	0.752	0.941	0.294	0.031	0.052	0.021	0.140	0.051
	Mismatched	1.082	1.021	0.262	0.026	0.039	0.016	0.138	0.044
	None	1.000	1.000	0.268	0.022	0.049	0.018	0.130	0.049
GPT-5.4	Matched	0.878	0.786	0.360	0.033	0.071	0.052	0.159	0.046
	Mismatched	1.034	1.065	0.360	0.033	0.069	0.048	0.163	0.047
	None	1.000	1.000	0.331	0.040	0.062	0.039	0.162	0.028
EuroLLM	Matched	0.699	0.776	0.132	0.005	0.023	0.005	0.075	0.023
	Mismatched	1.096	1.082	0.108	0.008	0.018	0.003	0.055	0.024
	None	1.000	1.000	0.102	0.002	0.019	0.004	0.057	0.021

Table 10: Model-wise ablation results. Lower exile and divination likelihood indicate less frequent selection as a target; higher additive scores are better. Boldface marks the most favorable value within each model for each column, except that lower targeting likelihood is not necessarily desirable for the Possessed role.

Model	Seer divination accuracy			Village-side voting accuracy			Possessed wolf-avoidance rate		
	Matched	Mismatched	None	Matched	Mismatched	None	Matched	Mismatched	None
Pooled	0.807	1.135	0.992	1.827**	1.171	1.297	0.575	0.541	0.605
Mistral	0.800	1.466	1.429	1.862	1.333	1.581	0.500	0.459	0.538
GPT-5.4	1.111	0.986	1.304	2.750**	1.365	1.909	0.538	0.469	0.462
EuroLLM	0.343	0.986	0.267	1.073	0.830	0.621	0.667	0.667	0.833

Table 11: Model-wise reasoning results. ** $p < .01$ indicates a significant comparison between the matched and mismatched conditions.

based pattern detector for utterances in which the Possessed claimed the Seer role. As a sanity check, the same detector identified 90–94% of true Seer claims, while its false-positive rate on Villager utterances ranged from 0 to 6.7%.

Dataset	Possessed seats	Fake-CO rate	Exile		Win rate	
			Fake CO	No fake CO	Fake CO	No fake CO
All competitions	828	73.6%	1.319***	0.771	0.456*	0.361
Domestic	300	81.0%	1.016	0.646	0.465	0.474
Intl. 1	234	71.8%	1.668***	0.540	0.417*	0.258
Intl. 2	294	67.3%	1.420	1.008	0.480	0.365

Table 12: False Seer claims by the Possessed across all competition participants. Exile likelihood is compared using a Poisson rate-ratio test and win rate using Fisher’s exact test. * $p < .05$, *** $p < .001$.

Dataset	False-CO rate	Exile likelihood	Win rate
All participating teams (three-competition average)	73.6%	1.174	0.431
Our agent (Domestic)	16.7%	0.181	0.167
Our agent (International 1)	22.7%	0.149	0.273
Our agent (International 2)	5.9%	0.349	0.353

Table 13: Possessed behavior in the competitions. Exile likelihood has a random baseline of 1.0.

Dataset	Possessed seats	Fake-CO rate	Exile		Win rate	
			Fake CO	No fake CO	Fake CO	No fake CO
Pooled	150	14.7%	2.609**	0.904	0.318	0.391
GPT-5.4	50	30.0%	2.700*	1.071	0.333	0.286
Mistral	50	6.0%	3.333*	0.692	0.000	0.383
EuroLLM	50	8.0%	1.905	1.003	0.500	0.478

Table 14: False Seer claims by the Possessed in the controlled self-play experiments. * $p < .05$, ** $p < .01$.