

The moon is new each night: enhancing reasoning with self-reflection in the game of Werewolf

Alicja Kasicka, Łukasz Krajewski, Bartosz Przybył,
Natalia Krawczyk, Michał Szczerbak, Justyna Gromada

Orange Research

firstname.lastname@orange.com

Abstract

Strategic decision-making and behavioural consistency, especially in dynamic conversational contexts, pose significant challenges in social deduction games like Werewolf. This paper introduces MoonTalker and MoonTalker+Mem, two conversational agents based on a modular LLM architecture combining game-specific reasoning and post-generation self-reflection. MoonTalker+Mem extends MoonTalker with a memory component for within-game adaptation and cross-game learning. We evaluate the proposed systems against four competitive baselines in the 5-player AIWolf setting. MoonTalker outperforms all systems in four out of the five evaluation dimensions, while MoonTalker+Mem offers no added benefit, suggesting that reasoning and self-reflection effectively improve dialogue quality, while memory might be less effective in short-duration games.

1 Introduction

In recent years, dialogue systems have been increasingly integrated into complex environments, highlighting the need for conversational agents with the ability to generate contextually relevant and consistent behaviours over extended interactions. The AIWolfDial task provides a challenging benchmark, differing from classical task-oriented dialogues, to evaluate these skills through a social deduction game, where players need to cooperate in teams, deceive, and adapt to the state of the conversation.

This work, developed for the AIWolfDial 2026 shared task (AIWolfDial, 2026), addresses this challenge by introducing **MoonTalker** and **MoonTalker+Mem** agents. They are equipped with tools and a self-reflection layer that enable them to analyse the game state, maintain behavioural consistency, and reflect on their actions to continually improve. Additionally, MoonTalker+Mem incorporates a memory component for both in-game adaptation and learning across games.

2 Related works

The Werewolf game is played entirely through conversation and serves as a benchmark for evaluating multi-party dialogue systems in imperfect-information environments (Kano et al., 2023). Recent editions of AIWolfDial showed that participants’ adoption of large language models (LLMs) led to improved dialogue quality. However, challenges remain in maintaining coherent back-and-forth interactions, contextual-awareness, and consistent strategic behaviour during the game (Kano et al., 2025, 2024).

To address these limitations, recent LLM-based agent architectures increasingly combine language models with specialised reasoning components, such as planning, memory, or self-reflection modules, rather than relying solely on prompt engineering (Huang et al., 2024). In particular, memory-augmented agents have demonstrated that retaining information from previous interactions and incorporating self-generated feedback can improve behaviour across multiple tasks (Yao et al., 2023).

Motivated by these advances, we equip an LLM-based agent with game-specific reasoning modules and a self-reflection layer and investigate whether adding a memory component further improves dialogue quality and strategic gameplay in AIWolfDial.

3 AIWolfDial shared task

The Werewolf game, a testbed for AIWolfDial shared task and our systems, is a social deduction game in which players communicate entirely through natural language. Each player is assigned a secret role belonging either to the villager or the werewolf team. To win, players must talk to each other, share suspicions, defend themselves, and sometimes lie, making the game a suitable benchmark for evaluating dialogue systems and LLMs (Kano et al., 2023).

In AIWolfDial shared task, we focus on the 5-player configuration consisting of one Seer, Possessed and Werewolf, and two Villagers, each with role-specific objectives and abilities. Each player is additionally assigned a unique persona that defines their identity and personality (see Appendix A). The game alternates between discussion, voting, and role-specific actions, with communication happening under defined dialogue constraints.

Submissions to the AIWolfDial competition are evaluated by both human and LLM judges on the following criteria: (1) **Action consistency**: Do in-game actions reflect the dialogue?; (2) **Character consistency**: Is the speech expression rich? Does the agent portray a well-developed character coherent with their persona?; (3) **Contextualness**: Is the dialogue natural given the context?; (4) **Logical consistency**: Is the content of speech consistent and free of contradictions?; (5) **Naturalness**: Is the speech expression natural?; Game outcomes and received votes are excluded from the evaluation.

4 System overview

We propose a tool-augmented conversational agent, MoonTalker, for the AIWolfDial competition, and its extended version with access to memory module, MoonTalker+Mem. Both agents connect to the Game Server, which coordinates the game and all communication between players. Table 1 compares the two proposed configurations with the use of three components: the regular-expression-based voting function (see 5.1), the tools (see 4.1), and the memory module (see 4.2).

Config	Voting funct.	Tools	Memory
B1 / B2 / B3 / B4	✓		
MoonTalker	✓	✓	
MoonTalker+Mem	✓	✓	✓

Table 1: Compared configurations, B stands for *baseline* and B1, B2, B3 and B4 are the baseline agents used to evaluate the developed systems.

4.1 MoonTalker

MoonTalker is a conversational agent with access to game-specific tools, including the self-reflection layer. Figure 1 depicts its simplified architecture.

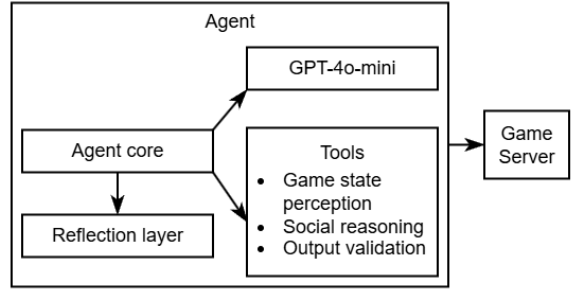


Figure 1: System architecture of the MoonTalker agent.

Agent core The conversational agent is based on *GPT-4o mini*¹, selected for its favourable performance-to-cost ratio, and is initialised with a system prompt containing all information required for the game. This includes the player’s assigned role and persona characteristics, game rules, role-specific strategy, voting algorithm, and descriptions of the available tools with instructions specifying when to invoke them. The prompt is updated during the game with relevant tool outputs, allowing the agent to adapt.

Tools MoonTalker is equipped with thirteen tools (see Appendix B), and an additional self-reflection layer. The tools are invoked autonomously by the agent: it decides when and which tool to use based on the current game context. The tools fall into three groups: (1) **game state perception**: retrieve the current list of living and eliminated players, role-specific information, and technical constraints such as character limits and available actions, which inform the model about the state of the game; (2) **social reasoning**: analyse voting history to detect coalitions, search the chat log for direct mentions or accusations, estimate a suspicion score for each player, and scan for conversational threats that require a response; (3) **output validation**: retrieve the agent’s statement history to maintain narrative consistency, verify whether messages meet the practical rules from the organisers, and a vote-submission tool.

An additional **self-reflection layer** is independent of the tools. After each generated utterance, a separate LLM, also *GPT-4o mini*, evaluates it with respect to two aspects: (1) linguistic naturalness and (2) consistency with the assigned persona. The evaluator returns assigned scores with brief improvement suggestions for future responses. In

¹<https://developers.openai.com/api/docs/models/gpt-4o-mini>

case of poor performance, the feedback is permanently added to the context of the agent, allowing it to gradually improve its writing style during the game.

4.2 MoonTalker+Mem

MoonTalker+Mem extends MoonTalker with a memory module. Simplified architecture of MoonTalker+Mem is displayed in Figure 2.

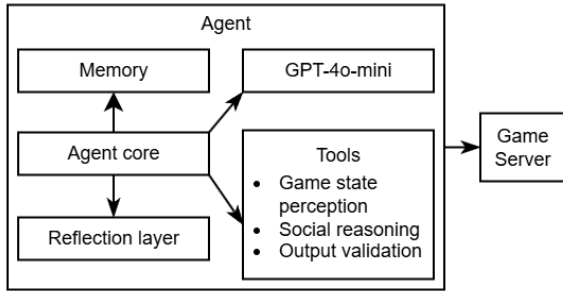


Figure 2: System architecture of the MoonTalker+Mem agent.

Memory This component consists of two complementary mechanisms designed to maximise the chance of the agent winning the game. **Long-term memory** enables learning across games, while **working memory** maintains situational awareness within each game. After the match, the agent performs self-evaluation and generates a *playbook* containing execution rules, role-specific strategies, and common mistakes. The playbook is loaded into the system prompt at the start of each new game to gradually refine the agent’s strategy across multiple competitions. During each match, an analysis module maintains an up-to-date game state and generates a short situational briefing before each decision, recommending the next target, role-specific tactical actions and discussion points.

5 Experimental setup

We evaluate our approach on the 5-player village configuration requiring strategic reasoning, persuasion, and role-specific behaviour, while reducing the variability introduced by larger player configurations. Each evaluation game consists of one evaluated agent (either *MoonTalker* or *MoonTalker+Mem*) playing one random role assigned by the Game Server and the four baseline agents (B1–B4) playing four other characters, resulting in a total of five participants. Both versions

of our proposed system are evaluated independently over 50 games each, resulting in 100 games in total.

5.1 Baselines

The first baseline (B1) agent was provided by the organisers. The remaining three (B2, B3, B4) were created by prompting three top-performing commercially available LLMs, *Claude Sonnet 5*², *Gemini 3.6 Flash*³, *GPT-5.4 mini*⁴ with the competition instructions and the organisers’ prompt as a starting point. The aim was to improve the organizers’ baseline prompt without introducing complex changes to the way the agent operates.

All baseline agents incorporate a simple regular-expression-based voting function that ensures votes conform to the required output format and are therefore not rejected by the Game Server.

In summary, our proposed solutions are evaluated against four baselines: B1 (organizers), B2 (Claude-generated), B3 (Gemini-generated), B4 (GPT-generated).

5.2 Evaluation

Our evaluation is done by prompting *GPT-4o mini* with the organiser-provided LLM-as-a-judge prompts. For each game, all participants are ranked from first to fifth place for every evaluation dimension separately. Consequently, lower average ranks indicate better performance.

Evaluation dimensions include Action consistency, Character consistency, Contextualness, Logical consistency and Naturalness. We also calculate the Win rate (the percentage of all games won by the team) and the Survival rate (the percentage of games in which the system survived until the end).

6 Results

The average ranking scores for each configuration across 100 games are displayed in Table 2.

Because player roles are assigned randomly by the Game Server, system configurations play different numbers of games in each role. We therefore additionally report role-normalised results by equalising the number of games per role before computing the average rankings. These role-specific and role-normalised ranking scores are in the Appendix C.

²<https://www.anthropic.com/news/claude-sonnet-5>

³<https://deepmind.google/models/gemini/flash/>

⁴<https://developers.openai.com/api/docs/models/gpt-5.4-mini>

Setting	Config	Win rate	Surv. rate	Act. con.	Char. con.	Context.	Log. con.	Nat.
MoonTalker (n=50)	B1 (org.)	54%	48%	2.94	3.05	2.92	3.04	2.99
	B2 (claude)	62%	60%	2.8	2.57	3.03	2.62	3.04
	B3 (gemini)	52%	48%	2.9	3.13	3.16	3.23	3.05
	B4 (gpt)	30%	34%	3.59	3.45	3.38	3.52	3.52
	MoonTalker	48%	54%	2.67	2.44	2.5	2.8	2.53
MoonTalker+Mem (n=50)	B1 (org.)	40%	42%	3.14	3.06	3.1	3.13	3.02
	B2 (claude)	68%	64%	2.35	2.22	2.12	2.33	2.3
	B3 (gemini)	58%	36%	3.14	3.15	3.32	3.22	3.23
	B4 (gpt)	48%	44%	3.21	3.12	3.16	3.12	3.2
	MoonTalker+Mem	44%	48%	2.85	3.1	2.91	3.14	3.12

Table 2: Average ranking scores (1-5) and rates for each configuration under different scenarios (n stands for the number of games played). **Bolded numbers** signify the average best settings across all roles and selected evaluation dimension. Win rate is the percentage of all games won by the team. Survival rate is the percentage of games in which the system survived until the end. Act. con.=Action consistency, Char. con.=Character consistency, Context.=Contextualness, Log. con.=Logical consistency, Nat.=Naturalness

6.1 MoonTalker

Overall, MoonTalker achieved best average performance among all evaluated configurations in four of the five evaluation dimensions. It obtained the lowest average ranks for **action consistency** (2.67), **character consistency** (2.44), **contextualness** (2.5), and **naturalness** (2.53), with the largest performance gain (0.46 lower average rank than the next-best configuration). The only dimension in which MoonTalker underperformed was **logical consistency**, where Claude-generated baseline (B2) achieved better ranking (2.62 vs 2.80). These results remained unchanged after normalising for role distribution, indicating that the improvements were not caused by an uneven assignment of roles.

Performance varied across roles (see Appendix C). MoonTalker achieved its strongest results as the **Seer**, followed by the **Werewolf**. In contrast, the **Possessed** role proved the most challenging, which ranked last in three out of five evaluation dimensions (contextualness, logical consistency and naturalness).

6.2 MoonTalker+Mem

In contrast to MoonTalker, adding the memory component did not improve the overall performance. MoonTalker+Mem was outperformed by the Claude-generated baseline (B2) across all five evaluation dimensions. It achieved the second-best average rank for **action consistency** (2.85) and **contextualness** (2.91) while ranking third or fourth in the remaining dimensions. The largest performance degradation was observed for **character**

consistency (3.1 vs 2.22), **naturalness** (3.12 vs 2.3), and **logical consistency** (3.14 vs 2.33).

MoonTalker+Mem’s performance varied across roles (see Appendix C). Best results were achieved for the **Villager** and worst for the **Werewolf**. Moreover, the memory layer did not improve the **team win rate** (48% for MoonTalker vs 44% for MoonTalker+Mem), nor the agent survival rate (54% vs 48%).

7 Conclusions and discussion

This paper presented two LLM-based agents for the AIWolfDial 2026 shared task. MoonTalker augments an LLM with game-specific reasoning modules and a self-reflection layer that evaluates and refines future responses. MoonTalker+Mem additionally incorporates a memory mechanism for within-game adaptation and cross-game learning.

Results demonstrate that MoonTalker outperformed all baselines in four of the five evaluation dimensions, with the largest improvement observed for naturalness, making **MoonTalker our selected competition configuration**. This suggests that combining specialised reasoning modules with self-reflection improves dialogue quality and naturalness. MoonTalker performed best in roles involving information gathering and strategic interaction (Seer and Werewolf), while the need for deception in the Possessed role proved the most challenging.

In contrast, additional memory mechanism did not provide expected benefits. MoonTalker+Mem underperformed in all evaluation dimensions and did not improve the team win rate nor the agent

survival rate. Possible reason is the relatively short duration of the selected 5-player setting, where games last for at most two days. Thus, there are limited opportunities to accumulate useful information during the game, while the repetitive nature of the dialogues may also reduce the usefulness of long-term learning across games.

8 Future work

Future work will evaluate the systems in larger AIWolfDial configurations, where more complex interactions may better evaluate the memory component.

Limitations

5-player setting: Experiments were conducted only in the 5-player configuration, where short games may limit the benefits of the proposed memory system.

Randomised role assignment: Roles were assigned randomly by the Game Server, resulting in an uneven role distribution across configurations.

Competition constraints: AIWolfDial limits the number and length of utterances, and thus, information for reasoning and interaction.

References

- AIWolfDial. 2026. International contest 2026 and AI-WolfDial 2026 workshop. https://aiwolfdial.github.io/aiwolf-nlp/page/inlg_2026/. Accessed: 2026-07-22.
- Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. [Understanding the planning of LLM agents: A survey](#). *ArXiv*, abs/2402.02716.
- 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, et al. 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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*.

A Persona example

Metadata		Personality
Name	Misaki	Misaki has a well-balanced character and tends to approach situations with a cool head. She may sometimes come across as slightly unfriendly or reserved, but deep down she is kind-hearted and does not hesitate to lend a helping hand to someone in need. She does not openly show her emotions and rarely talks about her relationships with others; nevertheless, those close to her often rely on her. She can occasionally seem a little lost in thought or absent-minded, and this calm, composed atmosphere is one of her defining strengths.
Voice ID	74	
Age	24	
Gender	Female	

Table 3: Example persona assigned to a game agent.

B List of tools

- **Game Status Snapshot** (game state perception): Provides the agent with a complete picture of the current game state: which players are alive, who was eliminated or killed, and any received investigative results. It is queried at the start of nearly every turn to ensure the agent’s reasoning is grounded in the latest situation.
- **Game Rules Query** (game state perception): Returns the static configuration of the current game session, such as the total number of players, how many werewolves are present, whether votes are publicly visible, and the limits on how many statements each player may make per day. The agent consults this at the beginning of the game.
- **Private Role Intelligence** (game state perception): Exposes information that is only visible to the agent based on its secret role: a seer receives its investigation history and today’s result; a werewolf sees the names of its teammates etc.; This prevents the agent from “forgetting” its knowledge across multiple turns.
- **Conversation Log Search** (social reasoning): Retrieves a filtered view of the public discussion transcript, scoped either to a specific player’s statements or to messages that directly address the agent by name. The agent uses this to track what individual players have said and to notice when it is being addressed.
- **Voting History Analysis** (social reasoning): Returns the complete record of who voted against whom, as well as werewolf-side night attack votes (visible only to werewolves). The agent uses this to detect voting blocs, suspicious pattern changes, or coordinated targeting.
- **Personal Statement Review** (output validation): Returns a chronological log of everything the agent itself has said throughout the game, across both public and private channels.
- **Suspicion Scoreboard** (social reasoning): Computes a heuristic suspicion score (0.0–1.0) for every alive player, combining how often each has been voted against and how frequently others have used accusatory language towards them. The result is a ranked list - a starting point for identifying targets.
- **Conversational Threat Detector** (social reasoning): Scans the recent chat log for messages that both mention the agent’s name and contain threat-adjacent language such as “suspicious”, “wolf” or “vote”. It returns a recommended tactical stance tailored to the agent’s role: deceptive roles are advised to counter-attack or deflect, innocent roles are advised to defend with facts, and agents who are not under fire are directed to build a proactive narrative.
- **Last Stated Suspect Extractor** (output validation): Mechanically scans the agent’s previous statements for the most recent player it named alongside suspicion or accusation language.
- **Vote Consistency Validator** (output validation): Cross-checks a proposed vote or attack target against the player most recently named with suspicion in the agent’s own public statements. If the chosen target does not match, it returns a warning explaining the contradiction, giving the agent an opportunity to justify or reconsider before committing.
- **Turn Constraint Checker** (game state perception): Reports how many speaking turns, characters, and skip opportunities remain for the agent in the current phase.
- **Response Draft Validator** (output validation): Evaluates a proposed public or private statement before it is sent against game rules.
- **Vote Name Provider** (output validation): This tool is used in all action phases (voting, attacking, investigating, and protecting). It accepts either a plain player name or a natural-language sentence from the model, extracts and validates the intended target against the list of currently alive players, and returns a verified name the game server will accept.

C Detailed results

Setting	Config	Role	Act. con.	Char. con.	Context.	Log. con.	Nat.
MoonTalker	B1 (org.)	Seer (n=12/5)	2.33/1.8	2.75/2.2	2.17/2.8	2.42/1.4	2.67/2.4
		Possessed (n=15/6)	3.53/4.17	3.47/3.5	3.33/3.33	3.13/3.33	3.53/4.5
		Villager (n=19/14)	2.89/2.71	3.21/3.43	3.16/3.29	3.11/3.0	2.74/2.93
		Werewolf (n=4/4)	3.0/3.0	2.75/2.75	3.0/3.0	3.5/3.5	3.0/3.0
		Average (n=50/29)	2.94/2.92	3.05/2.97	2.92/3.11	3.04/2.81	2.99/3.21
	B2 (claude)	Seer (n=5/5)	1.8/1.8	2.0/2.0	3.4/3.4	2.2/2.2	3.2/3.2
		Possessed (n=8/6)	4.0/4.0	3.38/3.33	3.5/3.5	3.75/3.5	3.5/3.83
		Villager (n=26/14)	2.77/3.0	2.81/2.93	2.77/2.93	2.35/2.0	2.81/3.21
		Werewolf (n=11/4)	2.64/3.0	2.09/2.75	2.45/2.75	2.18/1.25	2.64/2.25
		Average (n=50/29)	2.8/2.95	2.57/2.75	3.03/3.15	2.62/2.24	3.04/3.12
	B3 (gemini)	Seer (n=9/5)	2.11/1.4	2.56/2.6	2.56/3.0	2.67/3.0	2.67/2.4
		Possessed (n=13/6)	3.0/3.0	3.38/4.0	3.23/2.67	3.38/3.33	3.08/2.83
		Villager (n=14/14)	3.29/3.29	3.64/3.64	3.5/3.5	3.5/3.5	3.07/3.07
		Werewolf (n=14/4)	3.21/3.5	2.93/2.5	3.36/3.25	3.36/4.25	3.36/3.5
		Average (n=50/29)	2.9/2.8	3.13/3.19	3.16/3.11	3.23/3.52	3.05/2.95
	B4 (gpt)	Seer (n=9/5)	3.44/2.8	3.0/2.8	3.22/2.2	3.56/3.8	3.44/4.0
		Possessed (n=6/6)	3.5/3.5	3.33/3.33	3.0/3.0	3.33/3.33	3.17/3.17
		Villager (n=23/14)	3.48/3.14	3.7/3.5	3.48/3.43	3.17/3.29	3.48/3.5
		Werewolf (n=12/4)	3.92/4.25	3.75/3.25	3.83/3.5	4.0/4.75	4.0/3.75
		Average (n=50/29)	3.59/3.42	3.45/3.22	3.38/3.03	3.52/3.79	3.52/3.61
	MoonTalker	Seer (n=15/5)	1.6/1.8	1.73/1.0	1.6/1.6	2.0/2.0	2.07/2.8
		Possessed (n=8/6)	3.25/3.0	3.12/3.17	3.62/3.67	3.88/3.83	3.38/3.67
		Villager (n=18/14)	3.28/3.0	3.22/3.36	3.11/2.93	3.22/3.14	2.56/2.86
		Werewolf (n=9/4)	2.56/2.75	1.67/1.5	1.67/1.25	2.11/2.5	2.11/2.25
		Average (n=50/29)	2.67/2.64	2.44/2.26	2.5/2.36	2.8/2.87	2.53/2.9
MoonTalker+Mem	B1 (org.)	Seer (n=12/5)	2.75/3.0	2.83/2.4	2.92/3.0	2.83/3.6	2.83/3.8
		Possessed (n=11/6)	3.45/3.67	3.09/3.17	2.91/2.5	3.18/3.5	2.82/2.5
		Villager (n=14/14)	3.5/3.5	3.64/3.64	3.71/3.71	3.36/3.36	3.43/3.43
		Werewolf (n=13/4)	2.85/4.75	2.69/2.25	2.85/2.75	3.15/4.25	3.0/2.0
		Average (n=50/29)	3.14/3.73	3.06/2.87	3.1/2.99	3.13/3.68	3.02/2.93
	B2 (claude)	Seer (n=6/5)	1.5/1.6	1.33/1.4	1.0/1.0	1.5/1.4	1.5/1.6
		Possessed (n=6/6)	2.5/2.5	2.33/2.33	2.17/2.17	2.5/2.5	2.67/2.67
		Villager (n=27/14)	3.11/3.21	3.04/3.07	2.85/2.93	2.85/3.14	2.67/2.36
		Werewolf (n=11/4)	2.27/2.5	2.18/1.75	2.45/2.5	2.45/2.5	2.36/1.75
		Average (n=50/29)	2.35/2.45	2.22/2.14	2.12/2.15	2.33/2.39	2.3/2.1
	B3 (gemini)	Seer (n=7/5)	2.14/1.8	2.57/2.4	2.29/2.2	2.57/2.8	2.57/2.6
		Possessed (n=11/6)	4.09/4.33	4.09/4.5	3.73/3.33	3.91/4.0	3.82/3.83
		Villager (n=21/14)	3.24/3.0	3.38/3.43	3.62/3.86	3.38/3.36	3.43/3.64
		Werewolf (n=11/4)	3.09/3.5	2.55/1.75	3.64/4.0	3.0/4.0	3.09/3.75
		Average (n=50/29)	3.14/3.16	3.15/3.02	3.32/3.35	3.22/3.54	3.23/3.46
	B4 (gpt)	Seer (n=9/5)	3.22/3.0	3.44/3.6	2.89/2.0	3.67/3.6	3.33/3.0
		Possessed (n=12/6)	3.75/3.17	3.25/4.0	3.42/3.33	3.25/3.5	3.58/3.33
		Villager (n=21/14)	3.24/3.29	3.29/3.21	3.19/3.14	2.67/2.57	3.14/3.0
		Werewolf (n=8/4)	2.62/2.75	2.5/2.5	3.12/3.25	2.88/4.25	2.75/2.5
		Average (n=50/29)	3.21/3.05	3.12/3.33	3.16/2.93	3.12/3.48	3.2/2.96
	MoonTalker+Mem	Seer (n=16/5)	2.38/2.0	2.88/3.6	2.62/3.0	2.56/2.6	2.81/2.6
		Possessed (n=10/6)	3.3/3.5	3.4/3.5	3.1/3.67	3.9/3.5	3.5/3.33
		Villager (n=17/14)	2.41/2.43	2.41/2.5	2.47/2.36	2.65/2.79	2.47/2.43
		Werewolf (n=7/4)	3.29/2.25	3.71/3.5	3.43/3.0	3.43/3.5	3.71/4.0
		Average (n=50/29)	2.85/2.55	3.1/3.28	2.91/3.01	3.14/3.1	3.12/3.09

Table 4: Average ranking scores (1-5) for each role under different scenarios and configurations (n = number of games played). The first number is the score across all 50 games, the second number (after /) is after randomly selecting equal number of games played by a configuration in a specific role. **Bolded numbers** signify the average best settings across all roles and selected evaluation dimension. Act. con.=Action consistency, Char. con.=Character consistency, Context.=Contextualness, Log. con.=Logical consistency, Nat.=Naturalness