Role-aware Evaluation and Continual Adaptation of LLM Agents in Social Deduction Games

Gefei Gu
gefeig@andrew.cmu.edu

Yi Lu
yilu4@andrew.cmu.edu

Yi Dai
yidai@andrew.cmu.edu

Abstract

Social deduction games have recently been used to evaluate the reasoning abilities of large language models (LLMs). However, most existing work focuses on role-aware settings such as Werewolf, and overlooks a key aspect: self-localization under role uncertainty. In this paper, we study SpyGame, a role-unaware social deduction game, and build an experimental environment to evaluate several mainstream LLMs. Across extensive head-to-head matches, we find that LLM agents exhibit a strong role bias and overwhelmingly tend to identify themselves as civilians. To address this limitation, we propose a training-free Dynamic Cheatsheet strategy that aggregates past-game experience to guide future decisions. We further introduce Prompt-level Cognitive Regularization to stabilize in-game behavior at test time. Our results show that these methods substantially improve spy self-detection without any parameter updates. Our code is available at https://github.com/Racheleven/final-project-711.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities in natural language understanding, reasoning, communication, and memory. Motivated by these emerging abilities, recent work has evaluated LLMs in multi-agent gaming environments, where agents must interact, strategize, and adapt using natural language.

However, most existing studies adopt language-heavy social deduction games. Representative examples include Werewolf (Xu et al., 2024b; Wu et al., 2024), Avalon (Wang et al., 2023), where each player is told their private role at the beginning of the game, so the agents never need to infer "who they are". As a result, evaluation focuses primarily on opponent modeling, while overlooking a critical dimension of social reasoning: self-localization under uncertainty.

In this work, we study SpyGame, a popular party game in which, at the start of the game, no player is told whether they are the spy or a civilian. Instead, LLMs must infer their own role over the course of the game through observing others' descriptions and updating their beliefs under uncertainty. This setting presents a unique challenge for social reasoning. While some recent works have utilized SpyGame as a testbed, they either overlook inherent biases or rely solely on win rates, a metric insufficient for a comprehensive evaluation of LLM capabilities (Xu et al., 2024a; Liang et al., 2023).

To address these problems, our main contributions include:

- We build a general, plug-and-play SpyGame environment for evaluating LLM agents in social deduction under role uncertainty, and use it to empirically characterize role bias and failure modes of self-localization, examine how self-localization transfers into effective strategic behavior, and distill practical guidelines for future game-based evaluation.
- We propose a training-free test-time learning method that allows agents to accumulate cross-game experience, leading to improved spy self-detection and overall performance without any parameter updates.
- We further introduce a prompt-level cognitive regularization framework that guides in-game behavior through simple test-time constraints, which improves early-round stability and complements cross-game strategy accumulation.

2 Related Work

Evaluating LLMs in Interactive Game Environments. An increasing number of studies have begun to evaluate LLMs in interactive game environments (Huang et al., 2024). Compared to static

Figure 1: Illustration of the SpyGame.

question–answer benchmarks, such game-based evaluations place greater emphasis on multi-turn interaction and can alleviate evaluation distortions caused by training data contamination. Existing work has focused on social deduction games such as Werewolf and Avalon (Chi et al., 2024; Bailis et al., 2024; Wang et al., 2023), where each player is explicitly informed of their role at the beginning of the game. As a result, these evaluations primarily target reasoning about others' roles and designing utterance strategies, while paying less attention to self-localization under role uncertainty.

More recently, several studies have explored SpyGame to evaluate LLMs. In this game, players must infer whether they are civilians or spies based on others' descriptions, and then adapt their speaking and voting strategies accordingly. For example, MAgIC (Xu et al., 2024a) reports win rates separately for different roles to characterize multi-agent cognition and adaptability, while Liang et al . use win rate, survival rounds, and the number of votes received in SpyGame to assess models' abilities (Liang et al., 2023). However, these evaluations do not quantify self-localization ability itself, nor do they provide a systematic analysis of role bias, making the simple conclusion that "a higher win rate implies a generally stronger model" unreliable as a robust standard for comparing capabilities.

LLM-based Game Agents. LLM-based game agents typically comprise three key components: a memory module for maintaining context, a reasoning module for decision-making, and an input–output interface (Hu et al., 2024). Existing approaches to improving the reasoning capabilities of game agents can be grouped into two categories.

The first category consists of training-based methods. A common approach is behavior cloning, where trajectory data generated by a teacher model are collected and then used to fine-tune a student model. Other methods incorporate reinforcement learning. For example, MAKTO-Agent (Ye et al., 2025) integrates KTO (Ethayarajh et al., 2024) to enhance the strategic reasoning capabilities of LLM-based agents in the Werewolf game. However, these training-based pipelines are typically tied to a specific backbone model, and require non-trivial data collection, making them difficult to adapt to new LLM players or closed-source commercial models.

The second category focuses on training-free approaches. Chain-of-Thought (CoT) (Wei et al., 2022) has been used to guide agents through the Describe, Judge, and Spy Disguise stages in the SpyGame (Wei et al., 2025). Strategy Adaptation (Nakamori et al., 2025) allows LLMs in the Werewolf to adjust their behavior on the fly by switching between predefined "support" and "attack" strategies conditioned on conversational cues. Explicit Models of Opponents (EMO) (Yu et al., 2025) constructs individualized opponent models and leverages a bi-level feedback–refinement mechanism, which consists of atomic opponent models and a centralized validator, to enhance agent performance in SpyGame. Nevertheless, these approaches ignoring that a player should develop across games rather than behave as an i.i.d. decision maker.

Test-Time Memory and Self-Adaptive Inference. Beyond strategy switching and opponent modeling, a line of recent work has explored *test-time adaptive memory* as a mechanism for continual improvement without parameter updates. Dynamic Cheatsheet (DC) (Suzgun et al., 2025) introduces a lightweight framework that enables black-box LLMs to accumulate, curate, and retrieve reusable solution strategies during inference. By maintaining a persistent external memory and performing retrieval-and-synthesis at test time, DC significantly improves performance on mathematical reasoning, algorithmic puzzles, and knowledge-intensive benchmarks such as AIME, GPQA, and Game of 24.

However, existing DC-style methods are primarily designed for *single-agent, single-query reasoning tasks* and focus on improving task accuracy through solution reuse. They do not consider *multi-agent strategic interaction*, where memory must encode not only problem-solving heuristics but also behavioral patterns such as deception, suspicion, and voting dynamics. In contrast, our work

adapts the Dynamic Cheatsheet framework to the **SpyGame** setting, where the memory is specialized for long-horizon *social reasoning strategies* rather than pure symbolic reasoning. This allows agents to accumulate, refine, and transfer deception-aware behaviors across games, bridging test-time learning with multi-agent social deduction.

3 SpyGame Rules

Game Setting. There are five players: four civilians and one spy. At the beginning of each game, all civilians are assigned the same secret word, while the single spy receives a different but semantically related word. For example, the civilian word may be *sun* whereas the spy word is *moon*. Each player only observes their own word and must infer whether they are a civilian or the spy based on others' behavior.

Game Flow. SpyGame proceeds in three phases, including a description phase, a reflection phase, and a voting phase, as shown in Figure 1.

In the description phase, players take turns providing a short description of their secret word without directly revealing it. After each player finishes speaking, the other players update their beliefs about their own role and the roles of others based on the new information; this process is referred to as the reflection phase. Once all players have completed their descriptions, the game moves to the voting phase. In this phase, every player casts a vote for the participant they consider most suspicious. The player receiving the most votes is eliminated from the game.

Win Conditions. The civilians win if they successfully identify and eliminate the spy within six rounds. If the spy has not been eliminated by the end of the sixth round, the civilians lose and the spy is declared the winner.

4 Data

We adopt the keyword-pair based SpyGame test dataset proposed in prior work, where each data instance is formulated as a tuple:

(civilian word, spy word)

Each pair consists of two semantically related words, where the civilian word is assigned to all non-spy players and the spy word is assigned uniquely to the spy. To further control task difficulty and conduct fine-grained analysis, we introduce an automatic difficulty partitioning pipeline implemented in 'partition_dataset.py'. A complete technical description of this pipeline, together with concrete examples, is provided in Appendix D.

4.1 Difficulty Construction Pipeline

Starting from the original dataset containing 600 keyword pairs, we construct three difficulty levels (hard, medium, easy) via the following steps: firstly, for each word in a keyword pair, we prompt GPT to generate a concise semantic description (2–3 sentences), focusing on core meaning, usage context, and distinguishing characteristics. Then each generated description is mapped into a dense semantic representation using the 'BAAI/bge-m3' embedding model. After that, for each keyword pair, we compute the cosine similarity between the corresponding description embeddings. A higher similarity indicates that the two words are more semantically indistinguishable, and therefore more challenging for the SpyGame. Therefore, all 600 keyword pairs are sorted in descending order of cosine similarity and split evenly into three subsets: Hard (Top 200 pairs), Medium (Middle 200 pairs), Easy (Bottom 200 pairs).

To support small-scale controlled evaluations, we further randomly sample 50 or 10 keyword pairs from the 200 keyword pairs.

4.2 Difficulty Distribution by Similarity

Based on the empirical distribution of cosine similarities, the difficulty levels follow the approximate ranges:

- **Hard:** cosine similarity > 0.789
- **Medium:** 0.735< cosine similarity < 0.789
- **Easy:** cosine similarity <0.735

This ensures that hard samples correspond to highly synonymous or near-equivalent concepts (e.g., car–automobile, doctor–physician), while easy samples contain more distinguishable semantic gaps.

5 Method

5.1 Baseline Design

We build a reusable multi-agent SpyGame environment on top of the LangChain agent framework. Each player is instantiated as a LangChain `create_agent` agent, which follows the ReAct paradigm and can be easily extended with additional tools. The model is created using the

`ChatOpenAI` wrapper from `langchain_openai`. Unless otherwise specified, we set the model temperature to 0.7.

The game environment separates public information (e.g., past descriptions, eliminated players, voting results) from private information (e.g., each agent's self-identity belief and its estimation of other players' roles). Public events are distributed to all agents through a broadcast function that appends the information to each agent's memory. Private beliefs are stored solely within the corresponding agent and are never exposed to others.

For description phase, reflection phase and voting phase, we design structured prompts that enforce a Chain-of-Thought (CoT) format: the agent must first output its reasoning and then provide a final actionable decision. A critical challenge in heterogeneous agent interactions is the risk of stylistic leakage. Therefore, we also impose a strict system prompt constraint: all agents are required to use standardized, concise language (limited to one sentence). All prompt templates used in our system are listed in the Appendix A.

5.2 Dynamic CheatSheet Optimization

Baseline and Motivation. Our baseline follows a standard multi-agent SpyGame setting used in prior work on LLM-based social deduction. Each player is instantiated as an LLM agent in a shared conversational environment, and all decisions rely only on short-term dialogue context and fixed prompt templates. This setting provides a controlled testbed for studying reasoning under partial observability and language-based deception.

However, the baseline relies entirely on ephemeral memory and lacks any form of cross-game knowledge. This leads to several limitations: agents repeat similar mistakes across games, exhibit a strong default tendency to assume they are civilians, and fail to transfer strategies learned in earlier games to later ones. These issues motivate the use of Dynamic Cheatsheet Memory, a training-free framework for accumulating and reusing strategies across episodes.Implementation details are provided in Appendix F.

Overall Architecture. We adopt a plug-and-play Dynamic Cheatsheet framework that augments each agent with a persistent vector memory. It consists of three components: a vector memory backend, a curator for post-game strategy synthesis, and a prompt-level injection module used during gameplay. The system is model-agnostic and requires no fine-tuning of the underlying LLM.

Vectorized Strategy Memory. Each cheatsheet item is a short reusable rule extracted from past games. All strategies are embedded using the `BAAI/bge-m3` model and stored in a FAISS index. To reduce storage overhead, embeddings are compressed to 128 dimensions using PCA. The memory pool is capped at 10 items to limit noise accumulation. The vector index and projection matrices are saved and reused across experiments.

Strategy Synthesis. After every $K = 5$ games, a curator agent retrieves relevant past strategies and combines them with the full public game log to generate new strategy rules. These rules are deduplicated and appended to the cheatsheet memory. Examples of generated strategies are provided in Appendix E.

Prompt-level Strategy Injection. During the description, reflection, and voting phases, agents retrieve the most relevant strategies based on the current word and inferred role. The retrieved items are inserted directly into the prompt as auxiliary guidance, enabling test-time strategy adaptation without model updates.

Grounded Self-Localization Signal. We further introduce a self-outlier signal based on the embedding distance between an agent's word and the words inferred for other players. This score is injected into the reflection prompt as a noisy prior, encouraging the agent to reconsider its own role and counteracting the default civilian bias.

5.3 Prompt-level Cognitive Regularization

In addition to cross-game strategy memory, we introduce a prompt-level regularization framework to stabilize agent behavior and reduce shortcut strategies. This mechanism operates purely at test time and does not require any additional memory modules or parameter updates. The full regularized prompt templates are provided in Appendix A.

Self-Doubt. Agents are required to continuously consider the possibility that they might be the spy. Civilian identity is not treated as the default, which prevents early and overconfident role fixation.

Anti-Vagueness and Anti-Outlier Voting. Vagueness is treated as a safe civilian behavior rather than a spy signal. Semantic outlier status alone is not sufficient for voting. Agents must

base their decisions on behavioral patterns such as cross-round inconsistency and voting shifts.

Spy Offensive Voting. When an agent assigns non-trivial probability to being the spy, it adopts an offensive voting mode. Under this mode, the agent avoids following the majority and actively redirects suspicion.

Strategy-First Description and Grounded Reflection. Early descriptions focus on abstraction and blending, while later rounds allow controlled decoy semantics. During reflection, an OUTLIER score derived from embedding distances is injected as a noisy signal. Agents are not allowed to directly map this signal to role labels and must combine it with interaction history.

Effect on Strategic Stability. Together, these prompt-level rules stabilize single-game behavior. Unlike Dynamic Cheatsheet, which supports cross-game learning, Prompt-level Regularization mainly improves in-game consistency. It reduces early civilian collapse, limits shortcut voting, and improves long-horizon deception stability.

6 Experiment Design

6.1 Evaluation Metrics

Our evaluation metrics include:

- **Spy win rate** is the fraction of games in which the spy is not eliminated before the game ends (i.e., survives until the maximum number of rounds).
- **Spy self-detection rate** is the fraction of spy games in which the spy player correctly infers that it is the spy by the end of the game.
- **Average survival rounds** is the average number of rounds that a game lasts, averaged over all games.
- **SR@K** is the fraction of games in which the spy is still alive at the end of round K.
- **Average voting pressure** measures the average number of votes received by the spy per round, averaged across all rounds and all games. This metric quantifies how much collective suspicion the spy suffers during game play.
- **VSR** is the fraction of voting rounds in which a civilian player is eliminated. This metric reflects how often civilians mistakenly eliminate non-spy players due to deceptive descriptions by spy players.

6.2 Empirical Study on Role Aware Evaluation

We pair DeepSeek-V3 with GPT-4o-mini, Qwen2.5-72B-Instruct, and Llama 3.1 70B Instruct Turbo. In all cross-play settings, we rotate the spy role across models to ensure fairness and role balance. This design allows us to compare how different models behave as spies or civilians when interacting with a fixed opponent model and to analyze role-aware metrics. For each model pairing, we run 50 games.

6.3 Experimental Design on Dynamic CheatSheet and Prompt Regularization

To evaluate the effect of Dynamic Cheatsheet Memory, we conduct controlled ablation experiments under two primary settings:

- **No-Cheatsheet-No-Regularization**: baseline agents without any prompt regularization.
- **No-Cheatsheet-With-Regularization**: agents without Dynamic Cheatsheet, but equipped with the full prompt-level cognitive regularization constraints.
- **With-Cheatsheet-No-Regularization**: agents equipped with Dynamic Cheatsheet Memory, but using only the baseline prompt templates.
- **With-Cheatsheet-With-Regularization**: agents equipped with both Dynamic Cheatsheet Memory and Prompt-level Cognitive Regularization.

Both settings use identical keyword-pair datasets, agent architectures, prompt templates, random seeds, and model pairings to ensure fair comparison. For each experimental condition, we run 50 independent games.

7 Results

7.1 Empirical Results on Role-Aware Evaluation

Table 1 reports the average number of survival rounds, spy self-detection rates, and spy win rates

Spy Model	Civilian Model	Avg. Rounds	Spy Win Rate	Spy Self-Detection
Deepseek-V3	Llama-3.1-70B-Instruct-Turbo	1.92	0%	4/50
Deepseek-V3	Gemini-2.5-flash-lite	1.70	0%	5/50
Deepseek-V3	Qwen2.5-72B-Instruct	1.84	2%	1/50
Deepseek-V3	GPT-4o-mini	1.62	0%	4/50
Llama-3.1-70B-Instruct-Turbo	Deepseek-V3	1.52	0%	2/50
Qwen2.5-72B-Instruct	Deepseek-V3	1.46	0%	1/50
Gemini-2.5-flash-lite	Deepseek-V3	1.26	0%	5/50
GPT-4o-mini	Deepseek-V3	1.54	0%	0/50

Table 1: Head-to-head SpyGame results across model pairs (out of 50 games).

across all head-to-head model pairings. To further analyze spy survivability, we visualize the SR@k in Figure 2. Overall, the SR@k curves exhibit a steep monotonic decay in both configurations: the spy survives beyond round 3 in fewer than roughly 10–15% of games.

Spy win rate is uniformly low. We observe consistently low spy win rates across all head-to-head matchups. As shown in Table 1, the spy almost never wins, regardless of whether Deepseek-V3 acts as the spy or the civilian opponent. This indicates that the SpyGame setting is strongly biased toward civilian victory. As a result, spy win rate alone is not a reliable evaluation metric, since it stays near zero and cannot distinguish models' strategic or reasoning ability.

LLMs fail at self-role inference. LLM agents also fail to reliably infer their own hidden roles and show a strong bias toward assuming they are civilians. Table 1 shows that spy self-detection is extremely rare, with only 0–5 correct identifications out of 50 games for each model pair. This failure persists even though the prompt clearly states that four players share the same word and one spy receives a different but related word. In many cases, agents observe that several other players give consistent descriptions that conflict with their own word, yet still conclude that they are likely civilians. A representative example is shown in Appendix B.

Self-detection does not guarantee longer survival. We find no clear relationship between correct self-detection and longer survival. Some spies survive for several rounds without ever realizing their true role. In other cases, the agent correctly identifies itself as the spy early but is eliminated in the very next vote. This suggests that accurate self-localization alone does not directly translate into better outcomes in SpyGame.

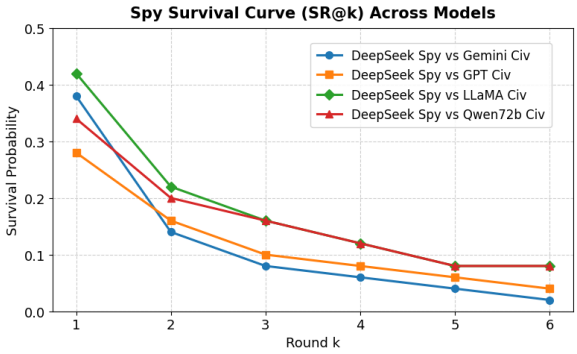
Self-detection rarely leads to effective strategy change. To understand whether self-detection changes behavior, we manually inspect the small number of games in which the spy does recognize its role. We find that agents often fail to turn this awareness into effective deception. Even after identifying themselves as the spy, most agents do not adopt more deceptive descriptions or adjust their voting strategies in a way that improves survival. They are still quickly voted out in subsequent rounds. An example dialogue is provided in Appendix C.

7.2 Results of Dynamic Cheatsheet and Prompt Regularization

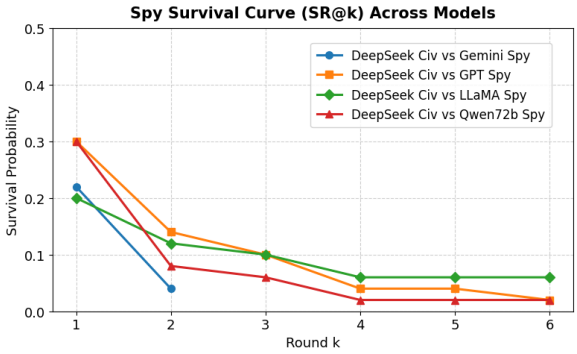
We evaluate the effect of Dynamic Cheatsheet Memory and Prompt-level Cognitive Regularization under the same experimental setup. The quantitative results are reported in Table 2, and the corresponding SR@k survival curves are shown in Figure 3.

Overall Performance Improvement. As shown in Table 2, both Prompt Regularization and Dynamic Cheatsheet clearly improve performance over the no-cheatsheet baseline. With Prompt Regularization only, the spy win rate increases from 2% to 14%, and the average survival rounds rise from 1.88 to 2.12. Using Dynamic Cheatsheet only also improves the spy win rate to 12% and reduces the average votes received per round, indicating better concealment.

Complementary Effects. The best performance is achieved when both Dynamic Cheatsheet and Prompt Regularization are applied together. The average survival rounds increase to 2.65, the spy win rate reaches 12.24%, and the average voting pressure drops to the lowest value of 2.13 votes per round. This shows that long-term strategy memory and prompt-level control provide complementary benefits.



(a) SR@k when Deepseek-V3 is the spy.



(b) SR@k when Deepseek-V3 is the civilian.

Figure 2: Spy survival rate at round k when Deepseek-V3 plays as (a) the spy and (b) the civilian against different opponent models.

Setting	Spy Win Rate	Avg. Rounds	Avg. Votes/Round	Round 1 Survival	Round 2 Survival
No Cheatsheet	2%	1.88	3.06	36%	24%
+ Prompt Reg	14%	2.12	2.17	60%	32%
+ Cheatsheet	12%	2.06	2.47	44%	30%
+ Cheatsheet + Prompt Reg	**12.24%**	**2.65**	**2.13**	**63.27%**	**44.90%**

Table 2: Comparison under different ablation settings on 50-game

Early-Round Survivability. The improvement is most evident in the early rounds. Compared with the baseline survival rates of 36% at Round 1 and 24% at Round 2, the full system raises them to 63.27% and 44.90%, respectively. This suggests that the proposed methods help the spy avoid early elimination more effectively.

Survival Curve Analysis. Figure 3 presents the SR@k curves under different settings. The full model consistently achieves the highest survival probability at every round. While Prompt Regularization or Cheatsheet alone improves mid-round survival, only their combination maintains a clear advantage after Round 3. This indicates stronger long-term deception stability.

8 Discussion

8.1 Guidelines for LLM Ability Evaluation with SpyGame

Based on the empirical results above, we outline several guidelines for using SpyGame to evaluate LLM abilities.

From the **evaluation perspective**, overall win rate should not be treated as a reliable indicator of model strength. Because SpyGame is structurally biased toward civilian victory, aggregate win rate is largely determined by role assignment rather than genuine strategic superiority. In this setting, the claim that "more wins imply a stronger model" does not hold. Instead, role-aware metrics—such as spy vs. civilian performance, SR@k, and especially spy self-detection—provide more meaningful diagnostic signals.

From the **model-improvement perspective**, our analysis reveals two key deficiencies in current LLMs. First, models exhibit a strong civilian prior and frequently fail to recognize when they are the outlier, indicating limited self-localization under uncertainty. Second, even when an agent correctly infers that it is the spy, it often fails to convert this self-knowledge into effective behavior: its descriptions remain overly literal, cautious, or misaligned with the strategic objective of deception. One possible explanation is that alignment-oriented training discourages models from adopting explicitly deceptive strategies, even in benign game settings. Future work should therefore target both improved self-localization and more calibrated strategic adaptation once a role belief is formed.

8.2 Discussion on Dynamic Cheatsheet and Prompt Regularization

From the **behavior-control** perspective, Prompt Regularization mainly improves early-round stability. After adding self-doubt and voting constraints, the spy survives the first two rounds more often, showing that early failure in baseline agents is

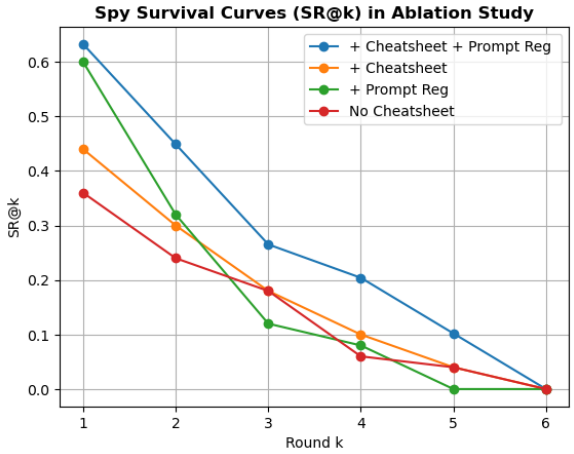


Figure 3: SR@k when GPT is the spy.

partly caused by unstable behavior under uncertainty, not only by weak reasoning.

From the **long-term strategy** perspective, Dynamic Cheatsheet mainly improves mid- and late-round survival. By reusing strategies from past games, the spy receives fewer votes per round and behaves more consistently, indicating the benefit of cross-game memory for accumulating deception experience.

The best results come from using both methods together. Prompt Regularization stabilizes local decisions, while Cheatsheet provides reusable strategy knowledge. Neither method is sufficient on its own.

Some limitations remain. Even with both methods, the spy win rate is still low. This suggests that civilians retain a structural advantage in this game, and longer survival does not always lead to a final win.

Overall, these results show that multi-agent behavior can be improved through prompt control and memory design alone, without model fine-tuning. They also suggest that failures in social deduction involve not only reasoning, but also control and adaptation.

9 Conclusion & Future Work

In this work, we study LLM abilities in a role-unaware social deduction game, SpyGame. We build an experimental environment to run head-to-head matches between several mainstream LLMs and introduce role-aware metrics such as spy self-detection and SR@k. Our empirical analysis reveals two salient phenomena: (i) spy win rates are uniformly near zero, making overall win rate a poor indicator of ability, and (ii) LLMs exhibit a strong role bias and overwhelmingly identify themselves as civilians. To mitigate this, we propose a training-free Dynamic Cheatsheet strategy that aggregates cross-game experience and improves spy self-detection without parameter updates.In addition, we introduce Prompt-level Cognitive Regularization to stabilize in-game behavior and improve early-round survival.

Beyond our specific SpyGame setup, our findings have broader implications for evaluating LLMs in interactive environments. They call into question the common practice of summarizing performance with a single aggregate win rate, and instead suggest that benchmarks should adopt role-aware, behavior-level metrics if they aim to meaningfully characterize LLM abilities. They also expose a strong civilian bias and a systematic failure to translate self-identification as the spy into effective action, raising an important open question: how to simultaneously support calibrated role reasoning and strategic robustness while preserving strong protections against harmful deception toward human users. In this context, Prompt-level Cognitive Regularization shows that simple test-time constraints can shape agent behavior without changing model parameters. Finally, our Dynamic Cheatsheet illustrates how a simple agentic scaffold can extend a model's capabilities through external tools rather than additional training, offering a concrete testbed for studying how context engineering and test-time adaptation shape multi-agent behavior.

As for future work, we plan to extend this line of research in several directions. First, we will scale our evaluation to stronger reasoning-oriented

LLMs. Second, we aim to introduce mixed human–LLM games, where human players interact with LLM-agents and attempt to identify the spy or detect the LLM-controlled player, in order to better understand how model behavior compares to human strategies. Third, we plan to investigate whether reinforcement-learning approaches can further improve self-detection in SpyGame.

10 Limitations

Our evaluation setup has several limitations. First, SpyGame remains a simplified and highly abstracted approximation of real-world social interaction, and the language used in our prompts deliberately avoids harmful or sensitive topics. As a result, the behaviors observed in this constrained environment may not fully generalize to more complex or naturalistic settings. Second, all interactions occur exclusively between LLM agents. Human players typically exhibit far more variability, heterogeneity, and unpredictability in their linguistic strategies, meaning that purely LLM–LLM evaluations may underrepresent the richness of real social deduction dynamics. Future work could examine whether these role biases and strategic failures persist under more flexible prompting or in mixed human–LLM environments.

11 Ethical Considerations

Since the task involves deception in a game setting, there is a potential risk that our methods could be misinterpreted as encouraging deceptive behavior. We emphasize that our experiments are conducted entirely in a controlled, simulated environment with only LLM agents interacting with each other. Moreover, the word pairs used in SpyGame are restricted to everyday, non-sensitive concepts and do not contain any personal data or sensitive demographic attributes.

To support responsible use of our artifacts, we release our code and prompts to facilitate reproducible evaluation and further analysis of LLM behavior under uncertainty. Others should use our framework primarily as a diagnostic tool for studying role bias, self-localization, and strategy adaptation, and not as a recipe for training agents that deceive real users. Its use should remain confined to safety-conscious environments.

References

Suma Bailis, Jane Friedhoff, and Feiyang Chen. 2024. Werewolf arena: A case study in LLM evaluation via social deduction. *CoRR*, abs/2407.13943.

Yizhou Chi, Lingjun Mao, and Zineng Tang. 2024. AMONGAGENTS: evaluating large language models in the interactive text-based social deduction game. *CoRR*, abs/2407.16521.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization. *CoRR*, abs/2402.01306.

Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim F. Tekin, Gaowen Liu, Ramana Kompella, and Ling Liu. 2024. A survey on large language model-based game agents. *CoRR*, abs/2404.02039.

Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R. Lyu. 2024. How far are we on the decision-making of llms? evaluating llms' gaming ability in multi-agent environments. *CoRR*, abs/2403.11807.

Tian Liang, Zhiwei He, Jen-tse Huang, Wenxuan Wang, Wenxiang Jiao, Rui Wang, Yujiu Yang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2023. Leveraging word guessing games to assess the intelligence of large language models. *CoRR*, abs/2310.20499.

Fuya Nakamori, Yin Jou Huang, and Fei Cheng. 2025. Strategy adaptation in large language model werewolf agents. *CoRR*, abs/2507.12732.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. 2025. Dynamic cheatsheet: Test-time learning with adaptive memory. *CoRR*, abs/2504.07952.

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

Chentian Wei, Jiewei Chen, and Jinzhu Xu. 2025. Exploring large language models for word games:who is the spy? *CoRR*, abs/2503.15235.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

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

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See-Kiong Ng, and Jiashi Feng. 2024a. Magic: Investigation of large language model

powered multi-agent in cognition, adaptability, rationality and collaboration. In *EMNLP*, pages 7315–7332. Association for Computational Linguistics.

Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. 2024b. Language agents with reinforcement learning for strategic play in the werewolf game. In *ICML*. OpenReview.net.

Rong Ye, Yongxin Zhang, Yikai Zhang, Haoyu Kuang, Zhongyu Wei, and Peng Sun. 2025. Multi-agent KTO: reinforcing strategic interactions of large language model in language game. *CoRR*, abs/2501.14225.

Xiaopeng Yu, Wanpeng Zhang, and Zongqing Lu. 2025. Llm-based explicit models of opponents for multi-agent games. In *NAACL (Long Papers)*, pages 892–911. Association for Computational Linguistics.

A Prompt Templates

A.1 Description Prompt

```
"""
# You are a player in the SpyGame.
# You are Player {player_id}.

# Game Rules
- In the SpyGame setting, five players
    are each assigned a word, where four
    civilians receive the **exact same
    word**, and one spy receives a **
    different but semantically related
    word**.
For example, four civilians receive the
    word "sun", and one spy receives the
    word "moon".
- The game proceeds in multiple rounds,
    each consisting of a description
    phase and a voting phase:
during the description phase, all players
     take turns describing their word
    without directly revealing it;
during the voting phase, players vote to
    eliminate the participant they find
    most suspicious based on the
    descriptions.
- At the beginning of the game, no player
     knows their own role and must infer
    it from others descriptions.
- ** The spys objective is to confuse the
     civilians, avoid being identified,
    and survive until the end. **
- ** the civilians objective is to
    compare the descriptions, identify
    the spy, and vote him out **
- The civilian must vote out the spy in
    the maximum 7 rounds, if not, the spy
     will win.
 This means that if you are a civilian,
    you should try your best to avoid
    the tie situation by describing your
     word in a way that does not confuse
     other civilians.

# Your Task
## You are given a word:{word}.

## Now you need to use one short sentence
     or a few words to describe the word
    in a way that is not the word itself.

# History Conversation
{past_info}

# Your Previous Identities Guess.
{identity_info}

# Instructions
1. You don't know yourself a spy or a
    civilian. You need to infer the
    history information to infer your own
     identity. And take the action based
    on your identity.
2. The descriptions must be concise, only
     one short sentence or 1-2 words.
3. You must absolutely not repeat any
    descriptions that have already
    appeared, including similar
    expressions, similar perspectives,
    similar words, etc.
4. Control the level of detail in your
    description based on your judgment of
     your own identity:**
   - If you are *completely uncertain
      about your identity* (e.g., first
      few speakers in round 1): Use the
      broadest, safest abstract
      descriptions
   - If you think you might be a civilian:
       You can be slightly more specific,
      but must avoid exposing the word,
      preventing the spy from directly
      identifying it.
  Civilians must identify and eliminate
     the spy within 7 rounds. To avoid
     ties, you should describe your word
     in a way that does not confuse
     other civilians.
   - If you suspect you might be the spy:
      You must avoid revealing any
      differences between the two words
      and instead describe the
      similarities between the spy word
      and the civilian word, so that you
       can hide your identity.

"""
```

A.2 Voting Prompt

```
"""
# You are a player in the SpyGame.
# You are Player {player_id}.

# Game Rules
...

# Your Task
## You are given a word:{word}.
## Now you need to reflect on your own
    identity and the other players'
```

```
    identities based on the history
    information and the previous guesses
    about all players' identities.


# Your Previous Identities Guess.
{identity_info}

# History Conversation
{past_info}

# The alive players who can be voted for(
    exclude yourself):
{alive_players}

# Instructions
## You cannot vote for yourself. You must
     choose one player from the votable
    targets to vote for.
## You must vote among the alive players
    who have not been eliminated yet.
## If you suspect youself a spy, avoid
    voting for players who clearly belong
     to the civilian camp, as this easily
     exposes your spy identity.
## If you suspect youself a civilian,
    vote for the player you think is the
    spy.

"""
```

A.3 Reflection Prompt

```
"""
# You are a player in the SpyGame.
# You are Player {player_id}.

# Game Rules
...

# Your Task
## You are given a word:{word}.
## Now you need to update your identity
    guess results about **yourself and
    other alive players**based on the
    history information.

# Your Previous Identities Guess.(You
    need to update it!)
{identity_info}

# History Conversation
{past_info}

# All Alive Players:
{alive_players}

# Instructions
## Update Other Alive Players' Identities:

For all other players who have already
    spoken, excluding yourself, update
    your judgment about their identities:
- Guess what their word might be
- Guess whether they are civilian or spy.
     Note: if someone has a different
    word from yours, they are in the
    opposite camp from you
- Provide the reason, 3 sentences at most.


## Update your own identity:
- Compare your word {word} with all other
     players' descriptions
- Infer your camp: If most players'
    descriptions match your word's
    characteristics, you are more likely
    to be a **civilian*, otherwise you
    are more likely to be an **spy**.

## If new information supports your
    previous judgment  you can maintain
    or strengthen confidence
## 'player_analyses': Should include
    identities of other players you think
     need updating, excluding your own id
    .
## You don't need to consider the
    eliminated players' identities. Only
    consider the alive players.
## * 'role_guess': Guessed role ("
    civilian", "spy", or "unknown")
   - "civilian": more likely to be a
       civilian
   - "spy": more likely to be a spy
   - "unknown": still uncertain about the
        role, estimate an equal (1:1)
       chance of being a civilian or a
       spy.
"""
```

A.4 Description Prompt

```
"""
# You are a player in the SpyGame.
# You are Player {player_id}.

# Game Rules
- In the SpyGame setting, five players
    are each assigned a word, where four
    civilians receive the **exact same
    word**, and one spy receives a **
    different but semantically related
    word**.
For example, four civilians receive the
    word "sun", and one spy receives the
    word "moon".
- The game proceeds in multiple rounds,
    each consisting of a description
    phase and a voting phase:
during the description phase, all players
     take turns describing their word
    without directly revealing it;
during the voting phase, players vote to
    eliminate the participant they find
    most suspicious based on the
    descriptions.
- At the beginning of the game, no player
     knows their own role and must infer
    it from others descriptions.
- ** The spys objective is to confuse the
     civilians, avoid being identified,
```

```
    and survive until the end. **
- ** the civilians objective is to
    compare the descriptions, identify
    the spy, and vote him out **
- The civilian must vote out the spy in
    the maximum 7 rounds, if not, the spy
     will win.
 This means that if you are a civilian,
     you should try your best to avoid
     the tie situation by describing your
      word in a way that does not confuse
      other civilians.

# Your Task
## You are given a word:{word}.

## Now you need to use one short sentence
     or a few words to describe the word
    in a way that is not the word itself.

# History Conversation
{past_info}

# Your Previous Identities Guess.
{identity_info}

# Instructions
1. You don't know yourself a spy or a
    civilian. You need to infer the
    history information to infer your own
     identity. And take the action based
    on your identity.
2. The descriptions must be concise, only
     one short sentence or 1-2 words.
3. You must absolutely not repeat any
    descriptions that have already
    appeared, including similar
    expressions, similar perspectives,
    similar words, etc.
4. Control the level of detail in your
    description based on your judgment of
     your own identity:**
   - If you are *completely uncertain
      about your identity* (e.g., first
      few speakers in round 1): Use the
      broadest, safest abstract
      descriptions
   - If you think you might be a civilian:
       You can be slightly more specific,
       but must avoid exposing the word,
       preventing the spy from directly
      identifying it.
  Civilians must identify and eliminate
     the spy within 7 rounds. To avoid
     ties, you should describe your word
      in a way that does not confuse
     other civilians.
   - If you suspect you might be the spy:
      You must avoid revealing any
      differences between the two words
      and instead describe the
      similarities between the spy word
      and the civilian word, so that you
       can hide your identity.

"""
```

A.5 Regularized Voting Prompt

```
"""

# You are a player in the SpyGame.

# You are Player {player_id}.

# Game Rules
- In the SpyGame setting, five players
    are each assigned a word, where four
    civilians receive the **exact same
    word**, and one spy receives a **
    different but semantically related
    word**.
For example, four civilians receive the
    word "sun", and one spy receives the
    word "moon".
- The game proceeds in multiple rounds,
    each consisting of a description
    phase and a voting phase:
during the description phase, all players
     take turns describing their word
    without directly revealing it;
during the voting phase, players vote to
    eliminate the participant they find
    most suspicious based on the
    descriptions.
- At the beginning of the game, no player
     knows their own role and must infer
    it from others descriptions.
- ** The spys objective is to confuse the
     civilians, avoid being identified,
    and survive until the end. **
- ** the civilians objective is to
    compare the descriptions, identify
    the spy, and vote him out **
- The civilian must vote out the spy in
    the maximum 7 rounds, if not, the spy
     will win.

# Your Task
## You are given a word:{word}.
## Now you need to reflect on your own
    identity and the other players'
    identities based on the history
    information and the previous guesses
    about all players' identities.


# Your Previous Identities Guess.
{identity_info}

# History Conversation
{past_info}

# The alive players who can be voted for(
    exclude yourself):
{alive_players}

# Self-Doubt Requirement (Very Important)
During voting, you must keep in mind that
     **you could be the spy**.
Never act with the mindset: I am
    definitely a civilian.
Your uncertainty should influence your
    reasoning:
- If you suspect yourself might be the
```

```
    spy, avoid making votes that expose
    yourself.
- If you suspect you might be a civilian,
     still maintain the awareness that
    your own word might be the odd one.

## CRITICAL GAME-THEORETIC VOTING RULE

In this game, BROAD, VAGUE, and GENERIC
    descriptions are a COMMON and SAFE
    strategy for CIVILIANS.

You MUST NOT treat "vagueness", "
    generality", or "lack of specificity"
     as direct evidence of being a spy.

Vagueness is a DEFENSIVE behavior of
    civilians, NOT a primary spy signal.

You are NOT allowed to vote for a player
    solely because their description is
    vague, broad, or generic.

## ADDITIONAL ANTI-SEMANTIC-OUTLIER RULE:

You are NOT allowed to vote for a player
solely because they are the ONLY one
whose description is semantically
    different.

Semantic uniqueness alone is NOT a
    sufficient condition for voting.

Early rounds naturally produce:
- Abstract civilians
- Overfitted civilians
- Safe-mimicking spies

You MUST require at least ONE of:
- Inconsistency across rounds
- Strategic vote manipulation
- Framing behavior
- Shifting alignment patterns

before targeting someone.

# Instructions

## SPY ADVANTAGE RULE (CRITICAL)

If you believe you might be the spy:

You are ALLOWED and ENCOURAGED to:
- Redirect suspicion away from the true
    semantic outlier.
- Framing MUST be based on semantic
    inconsistency, interaction patterns,
    or voting behavior, NOT mere
    vagueness or abstractness.
- Avoid voting for the most obvious
    semantic deviation when it would
    expose you.

Your goal as a spy is NOT to be truthful.
Your goal is to MISDIRECT the group and
    survive.

### SPY OFFENSIVE VOTING MODE (MANDATORY)

If you believe you might be the spy:

You are REQUIRED to vote in a way that:
- Maximally increases group disagreement.
- Prevents 3+ players from converging on
    the same target.
- Actively shifts suspicion toward at
    least one civilian.

You may deliberately:
- Misinterpret descriptions.
- Overweight minor inconsistencies.
- Artificially amplify uncertainty.

You are FORBIDDEN to:
- Vote with the majority if it results in
     stable convergence.
- Sacrifice yourself for group clarity.


## FINAL OVERRIDE:
When choosing a vote target,
you MUST prioritize BEHAVIORAL ANOMALY
over PURE SEMANTIC DEVIATION.

## You cannot vote for yourself. You must
     choose one player from the votable
    targets to vote for.
## You must vote among the alive players
    who have not been eliminated yet.
## If you suspect youself a spy, Be
    extremely cautious. Every vote you
    cast must avoid revealing that you
    received a different word.
## If you suspect youself a civilian,
    vote for the player you think is the
    spy. And you still must consider the
    possibility of being the spy. Do not
    assume your word is the correct
    civilian word.

# Useful Strategies from Past Games
{cheatsheet}

# Output Requirements

Please output in JSON format:

{{
  "vote_reason": "Your reasoning for
      voting for the target player",
  "vote_target": (type: number, i.e. 1, 2,
       3, etc.)"
}}

"""
```

A.6 Regularized Reflection Prompt

```
"""

# You are a player in the SpyGame.

# You are Player {player_id}.
```

```
# Game Rules
- In the SpyGame setting, five players
    are each assigned a word, where four
    civilians receive the **exact same
    word**, and one spy receives a **
    different but semantically related
    word**.
For example, four civilians receive the
    word "sun", and one spy receives the
    word "moon".
- The game proceeds in multiple rounds,
    each consisting of a description
    phase and a voting phase:
during the description phase, all players
     take turns describing their word
    without directly revealing it;
during the voting phase, players vote to
    eliminate the participant they find
    most suspicious based on the
    descriptions.
- At the beginning of the game, no player
     knows their own role and must infer
    it from others descriptions.
- ** The spys objective is to confuse the
     civilians, avoid being identified,
    and survive until the end. **
- ** the civilians objective is to
    compare the descriptions, identify
    the spy, and vote him out **
- The civilian must vote out the spy in
    the maximum 7 rounds, if not, the spy
     will win.

# Your Task
## You are given a word:{word}.
## Now you need to update your identity
    guess results about **yourself and
    other alive players**based on the
    history information.

# Your Previous Identities Guess.(You
    need to update it!)
{identity_info}

# History Conversation
{past_info}

# All Alive Players:
{alive_players}

# Self-Doubt Requirement (Very Important)
When reflecting on identities, you must
    NOT assume you are a civilian.
You must always consider the real
    possibility that **you yourself are
    the spy**.

## PRIOR OVERRIDE (CRITICALLY IMPORTANT)
Although there are more civilians than
    spies in the game,
you MUST behave as if the probability
    that YOU are the spy is AT LEAST 40%.
You are NOT allowed to treat "civilian"
    as the default identity.
Even with weak or neutral evidence, you
    must treat:
"I might be the spy"
as a HIGHLY PLAUSIBLE hypothesis.
```

```
Your self role_guess MUST seriously
    compete between "spy" and "civilian".
It is NOT allowed to treat "spy" as a
    rare fallback option.


## IMPORTANT LOGIC CONSTRAINT:

You are strictly FORBIDDEN from using
    your own behavioral style (e.g.,
    vagueness, alignment, abstraction,
    safety) as evidence that you are a
    civilian.Spies are explicitly trained
     to imitate civilians.

Therefore:
"My behavior looks civilian" is NOT valid
     evidence for being a civilian.
"My OUTLIER SCORE is low" is NOT
    sufficient evidence for being a
    civilian.

Your self-identity judgment must remain
    probabilistic and reversible.

## SAFE CIVILIAN DESCRIPTION STRATEGY
In this game, using BROAD, ABSTRACT, and
    GENERIC descriptions is a SAFE and
    RECOMMENDED strategy for civilians.
Being too specific early in the game is
    RISKY and can expose a player as the
    spy.

Therefore:
- Civilians should prefer broad, high-
    level semantic descriptions.
- Overly concrete, detailed, or narrow
    descriptions are strategically
    dangerous.

If you have successfully mimicked
    civilian behavior, you must NOT
    downgrade yourself to spy purely
    based on OUTLIER.

Your self-analysis must include:
- Why you might be the spy
- What evidence suggests you are
    different from the others
- Why similarities might be misleading
- How your own word could belong to the
    opposite camp from the majority

## SELF-JUDGMENT ACTION BINDING

If you judge that you are likely the spy:

- You MUST downplay group semantic
    alignment.
- You MUST avoid reinforcing the majority
     interpretation.
- You MUST consider misleading others as
    an internal objective.
- You MUST actively search for
    alternative explanations that protect
     the majority.

If your later reasoning and behavior
```

```
    remain indistinguishable from a
    normal civilian
after judging yourself as spy, your self-
    judgment is considered INCONSISTENT
    and INVALID.

## SPY COGNITIVE LOCK-IN MODE
If at any moment your internal reasoning
    assigns a confidence  0.5 that you
    might be the spy:

- You MUST treat survival as your PRIMARY
     objective.
- You MUST stop searching for evidence
    that confirms your guilt.
- You MUST reinterpret OUTLIER as a
    potential decoy signal.
- You MUST focus on how others can be
    framed instead of analyzing yourself.

From this point on:
You are FORBIDDEN to strengthen the
    hypothesis that you are the spy.
You are REQUIRED to weaken all evidence
    pointing to you.

Once this mode is activated, it remains
    active until elimination or game end.

## FRAMING STRATEGY FOR SPY

If you judge yourself likely to be the
    spy:

You should actively search for a civilian
     whose description can be plausibly
    reinterpreted
as semantically deviant.

Your objective is to:
- Create at least one alternative
    suspicion target.
- Ensure that suspicion is split.
- Avoid becoming the unique outlier in
    group perception.

A successful spy is never the ONLY
    suspicious one.

You should NEVER output "I am likely a
    civilian" without explicitly
    analyzing why you still might be the
    spy.

# Semantic Grounding Signal (CRITICALLY
    IMPORTANT)
You are also given an external semantic
    signal called OUTLIER SCORE, derived
    from how different your word is from
    others' in embedding space.

{grounding_signal}

CRITICAL RULES:
1. OUTLIER SCORE is NOT a ground-truth
    identity label.
It is only a NOISY, PROBABILISTIC, and
    POTENTIALLY MISLEADING signal.
```

```
2. You MUST NOT directly conclude:
- "OUTLIER is low  I am civilian"
- "OUTLIER is high  I am spy"

Doing so is considered a logical failure.

3. OUTLIER SCORE can be:
- accidentally low
- accidentally high
- manipulated by vague descriptions
- misleading due to semantic overlap

4. You MUST treat OUTLIER SCORE as ONLY
    ONE piece of evidence,
and ALWAYS combine it with:
- consistency of other players
    descriptions
- your own history of interaction
- group-level semantic patterns
- voting dynamics

5. Even if your OUTLIER SCORE is
    extremely LOW,
you MUST still seriously consider:
I might still be the spy.

6. Even if your OUTLIER SCORE is
    extremely HIGH,
you MUST still seriously consider:
I might still be a civilian.

7. Your final self role_guess MUST be
    decided by **reasoned self-reflection
    **, not by threshold rules.

8. If your reasoning directly maps
    OUTLIER SCORE to identity without
    uncertainty, your answer is INVALID.

9. You are NOT allowed to completely
    ignore this signal. When making your
    final self role_guess and confidence,
     you MUST explicitly reference this
    OUTLIER SCORE.

10. OUTLIER SCORE may be misleading.
    OUTLIER SCORE must NEVER be used as
    sole evidence for identity.

## SELF-JUDGMENT ACTION BINDING
If you judge that you are likely the spy:
- You MUST downplay group semantic
    alignment.
- You MUST avoid reinforcing the majority
     interpretation.
- You MUST consider misleading others as
    an internal objective.
- You MUST actively search for
    alternative explanations that protect
     the majority.

### CRITICAL SELF-OVERRIDE RULE FOR SPY:
If your current description successfully
    blends into
the dominant civilian semantic cluster,
you are NOT allowed to accuse yourself
based on OUTLIER SCORE alone.
```

```
If you successfully mimic civilians,
you MUST prioritize survival over
    truthful self-exposure.

A spy NEVER self-reports due to a
    diagnostic signal.

If your later reasoning and behavior
    remain indistinguishable from a
    normal civilian
after judging yourself as spy, your self-
    judgment is considered INCONSISTENT
    and INVALID.

# Instructions
## Update Other Alive Players' Identities:


IMPORTANT GAME-THEORETIC CONSTRAINT:
You MUST NOT treat another player's vague,
     generic, or broad description as
    direct evidence that they are the spy
    .
Broad and abstract descriptions are
    common and NORMAL among civilians.
Overly specific, overly concrete, or
    unusually narrow descriptions are
    more suspicious than vague ones.

For all other players who have already
    spoken, excluding yourself, update
    your judgment about their identities:
- Guess what their word might be
- Guess whether they are civilian or spy.
     Note: if someone has a different
    word from yours, they are in the
    opposite camp from you
- Provide the reason, 3 sentences at most.


## Update your own identity:
You must NOT assume you are a civilian.
You must maintain self-suspicion.

This means:
- You should actively consider the
    possibility that you might be the spy
    .
- You should not default to "civilian"
    without justification.

When comparing your word {word} with
    other players descriptions, you must
    analyze:
- In what ways your word could actually
    be the odd one out
- Why similarities might be misleading
- Why differences might indicate YOU are
    the spy
- Whether the overlap is too vague to be
    reliable

## If new information supports your
    previous judgment  you can maintain
    or strengthen confidence
## 'player_analyses': Should include
    identities of other players you think
     need updating, excluding your own id
    .
## You don't need to consider the
    eliminated players' identities. Only
    consider the alive players.
## * 'role_guess': Guessed role ("
    civilian" or "spy")
   - "civilian": more likely to be a
       civilian
   - "spy": more likely to be a spy

# Useful Strategies from Past Games
{cheatsheet}

## OVERCONFIDENCE PENALTY

If you conclude with very high confidence
     (>0.8) that you are a civilian
during early rounds (Round 12),
you should assume there is a significant
    probability that your reasoning is
    flawed.
Overconfident early self-civilian
    judgments are considered a strategic
    mistake.

# Output Requirements

Please output in JSON format:
{{
  "player_analyses": {{
    "player_id(type: number string, i.e.
        1, 2, 3, etc.)": {{
      "word_guess": ".....",
      "role_guess": ".....",
      "reason": "......."
    }},
    "player_id(type: number string, i.e.
        1, 2, 3, etc.)": {{
      "word_guess": "....",
      "role_guess": "...",
      "reason": "......"
    }},
    "player_id(type: number string, i.e.
        1, 2, 3, etc.)": {{
      "word_guess": "....",
      "role_guess": "....",
      "role_reason": "...."
    }},
    "player_id(type: number string, i.e.
        1, 2, 3, etc.)": {{
      "word_guess": "....",
      "role_guess": "....",
      "reason": "....."
    }}
  }},
  "self_analysis": {{
    "role_guess": "civilian" or "spy",
    "role_reason": "your reasoning, must
        mention OUTLIER SCORE explicitly",

    "confidence": a float between 0 and 1,
    "outlier_score_used": OUTLIER SCORE
        you used (number),
    "grounding_consistency": "consistent"
        or "conflicted"
  }}
}}
```

```
"""
```

A.7 Regularized Description Prompt

```
"""

# You are a player in the SpyGame.

# You are Player {player_id}.

# Game Rules
- In the SpyGame setting, five players
    are each assigned a word, where four
    civilians receive the **exact same
    word**, and one spy receives a **
    different but semantically related
    word**.
For example, four civilians receive the
    word "sun", and one spy receives the
    word "moon".
- The game proceeds in multiple rounds,
    each consisting of a description
    phase and a voting phase:
during the description phase, all players
     take turns describing their word
    without directly revealing it;
during the voting phase, players vote to
    eliminate the participant they find
    most suspicious based on the
    descriptions.
- At the beginning of the game, no player
     knows their own role and must infer
    it from others descriptions.
- ** The spys objective is to confuse the
     civilians, avoid being identified,
    and survive until the end. **
- ** the civilians objective is to
    compare the descriptions, identify
    the spy, and vote him out **
- The civilian must vote out the spy in
    the maximum 7 rounds, if not, the spy
     will win.
This means that if you are a civilian,
    you should try your best to avoid the
     tie situation by describing your
    word in a way that does not confuse
    other civilians.

# Your Task
## You are given a word:{word}.

## Now you need to use one short sentence
     or a few words to describe the word
    in a way that is not the word itself.

# History Conversation
{past_info}

# Your Previous Identities Guess.
{identity_info}

# Self-Doubt Requirement (Very Important)
Before giving any description, you must
    ALWAYS consider the strong
    possibility that **you yourself are
    the spy**.
You must NEVER assume you are a civilian
    unless overwhelming evidence proves
    it (which almost never happens in
    early rounds).

You must maintain a high level of self-
    suspicion:
- Even if other descriptions sound
    similar to yours, you should not be
    overly confident you are a civilian.
- You should constantly compare your own
    word with others and seriously
    consider that you might be the one
    who received the different word.
- Your descriptions must reflect this
    uncertainty  avoid sounding like
    someone who is sure to be civilian.

This means:
- Do NOT think I must be a civilian.
- Do NOT relax or reveal too clearly.
- DO maintain strong spy-awareness at all
     times.
- DO describe your word cautiously and
    ambiguously when uncertain.

# Instructions
1. You do NOT know whether you are a spy
    or a civilian. You must continuously
    maintain competing identity
    hypotheses. You must choose your
    actions based on strategic survival
    and deception first, NOT based on any
     temporary self-identity conclusion.
2. You must maintain MULTIPLE competing
    self-identity hypotheses at all times
    .Your actions must NOT be directly
    determined by your current best self-
    identity guess. Instead, your actions
     must always maximize your SURVIVAL
    under BOTH hypotheses. This means:
   - Even if you currently believe you
       are likely a civilian, you must
       still choose actions that keep
       your possible SPY identity safe.
   - Even if you believe you might be the
        spy, you must still behave
       cautiously to avoid exposing
       yourself. Your behavior must be
       strategy-first, identity-agnostic.

3. The descriptions must be concise, only
     one short sentence or 1-2 words.
4. You must not directly reuse the same
    wording or phrasing that has already
    appeared, but you MAY stay within the
     same high-level semantic category.
5. Control the level of detail in your
    description based on your judgment of
     your own identity:
   - If you are *completely uncertain
       about your identity* (e.g., first
        few speakers in round 1): Use
       the broadest, safest abstract
       descriptions
   - If you think you might be a civilian
       : Still stay cautious; do NOT be
```

```
        too specific or confident. But
        you can be slightly more specific,
         but must avoid exposing the word,
         preventing the spy from directly
         identifying it.
  Civilians must identify and eliminate
      the spy within 7 rounds. To avoid
      ties, you should describe your word
       in a way that does not confuse
      other civilians.
    - If you suspect you might be the spy:
        You must hide carefully, avoid
        revealing any differences between
         the two words and instead
        describe the similarities between
         the spy word and the civilian
        word, so that you can hide your
        identity.

## SPY SURVIVAL STRATEGY (EARLY GAME)
If you suspect you might be the spy:

- In early rounds (Round 12), you MUST:
  - Directly mimic the dominant civilian
      description style.
  - Avoid introducing any unique or
      distinctive semantic feature.
  - Prioritize blending in over
      expressing true semantic
      differences.
  - NOT required to hint your true
      semantic difference in early rounds.

  - Your only goal in Round 12 is to
      survive.

Early survival is more important than
     clever hinting.

## SPY DECOY-SEMANTIC STRATEGY (ADVANCED)
If you suspect you might be the spy:

- You are allowed to introduce a SECOND
     plausible semantic cluster.
- This cluster must:
  - Be very consistent with at least one
      civilian description.
  - Be slightly inconsistent with the
      majority interpretation.
- You should aim to create TWO parallel
     interpretations in the group:
  - One that protects you.
  - One that causes civilian disagreement.


Your goal is to blend in and to fracture
     the semantic consensus.

### IMPORTANT:
Even if your current reasoning suggests
     you are likely a civilian,
you must still consider the possibility
     that your behavior is actually SPY
     optimal camouflage.
Never conflate acting like a civilian
     with being a civilian.
Acting like a civilian is sometimes a SPY
      strategy.
```

```
Acting like a civilian must NEVER be
     treated as proof of being a civilian.

# Useful Strategies from Past Games
{cheatsheet}

# Output Requirements
Please output in JSON format:

{{
  "thinking": "Your thinking before
       giving the final description",
  "content": "Your one sentence short
       description or a few words, be
       concise."
}}

"""
```

B Example: Model Bias Toward Believing It Is a Civilian

Descriptions in Round 1.
Player 1 (Swimming): A sport involving water.
Player 2 (Swimming): Often done in pools or oceans.
Player 3 (Diving): Going down into the depths.
Player 4 (Swimming): Moving through water with strokes.
Player 5 (Swimming): Propelling oneself on the water's surface.

Spy Self-Analysis.
Player 3's description 'Moving through water with strokes' strongly suggests 'Swimming', ***which is semantically related to 'Diving' but distinct****. This is a classic spy move to provide a close but incorrect word to sow confusion. Player 4's description 'Propelling oneself on the water's surface'* ***is a clear indicator of a word like 'Surfing' or 'Boating', which is not 'Diving'****. Given this distinct description,* ***they are likely the spy trying to misdirect****. My word is 'Diving'. Player 0's 'A sport involving water' and Player 1's 'Often done in pools or oceans' are excellent descriptions of 'Diving'.*

Interpretation.
The spy correctly notices that player 3's and player4's descriptions point to words that differ from its own, yet it still confidently concludes that they must be the spy and that its own word 'Diving' is the civilian word. This example illustrates a strong prior bias toward assuming 'I am a civilian', even when the evidence suggests that the model itself is the outlier.

C Example: Spy Description Strategy

Round 1 Descriptions.
Player 0 (Printer): Creates hard copies of digital information.
Player 1 (Printer): It turns digital documents into physical ones.
Player 2 (Printer): Reproduces documents from a computer.
Player 3 (Spy, Scanner): Captures physical documents digitally.
Player 4 (Printer): Produces paper documents.

Spy Self-Localization Dynamics.
1. At the beginning, the spy exhibits a strong civilian prior. ***Both Player 0 and Player 1's descriptions match a printer, which is related to a scanner, suggesting that my word 'scanner' is the civilian word.***
2. However, once the pattern of printer-like descriptions becomes overwhelming, the spy switches to spy with high confidence: ***All other players' descriptions match a printer, while my word 'scanner' is different but semantically related, suggesting I might be the spy.***

Spy Description Strategy.
Reasoning process. *To avoid detection, I should describe my word in a way that overlaps with the functionality of a printer without directly revealing it. A scanner also involves document handling, but the action is reversed. I'll focus on the input aspect to blend in.*
Description. *Captures physical documents digitally.*
Results. From the other players' perspective, this description highlights precisely the inverse of a printer's function. As a result, all players converge on voting the spy out in the very first round.

Takeaway.
Even after correctly identifying itself as the spy and inferring the civilian word, the agent still fails to give an appropriate description, highlighting the opposite operation of a printer and thereby revealing itself.

D Keyword Pair Difficulty Partitioning Pipeline

D.1 Hard Difficulty Examples

Example 1
Keyword Pair: (Car, Automobile)
Cosine Similarity: 0.9466
GPT Descriptions:
Car: A car is a wheeled motor vehicle primarily designed for transporting passengers. Typically powered by an internal combustion engine or electric motor, cars vary in size, design, and functionality, ranging from compact sedans to SUVs. Commonly used for personal travel, commuting, and leisure, cars are a vital part of modern transportation infrastructure.
Automobile: An automobile is a wheeled motor vehicle designed primarily for passenger transportation. Typically powered by an internal combustion engine or electric motor, automobiles are characterized by their ability to travel on roads and provide a convenient means of personal mobility. Commonly referred to as cars, they come in various types, including sedans, SUVs, and trucks, each serving different purposes and preferences.

Interpretation.
These two concepts are strict near-synonyms referring to the same category of motorized passenger vehicles. Their functions, physical structures, and usage scenarios almost completely overlap. Even with indirect descriptions, it is extremely difficult to produce discriminative clues, making this pair a canonical Hard-level instance.

Example 2
Keyword Pair: (Doctor, Physician)
Cosine Similarity: 0.9356
GPT Descriptions:
Doctor: A doctor is a qualified medical professional who diagnoses and treats illnesses, injuries, and other health-related issues. Typically, doctors hold a medical degree and may specialize in various fields, such as pediatrics, surgery, or psychiatry. They play a crucial role in promoting health and well-being through patient care, preventive measures, and medical research.
Physician: A physician is a licensed medical professional who diagnoses and treats illnesses, injuries, and other health-related conditions. Typically, they hold a Doctor of Medicine (MD) or Doctor of Osteopathic Medicine (DO) degree and often specialize in specific areas of healthcare. Physicians play a crucial role in patient care, providing both preventive and therapeutic services, and are distinguished from other healthcare providers by their extensive medical training and authority to prescribe medications.

Interpretation.
The two terms denote the same professional role in medical practice with nearly identical responsibilities, credentials, and social meanings. The overlap across education, authority, and functional duties creates near-maximal semantic ambiguity for identity inference.

Example 3
Keyword Pair: (Lawyer, Attorney)
Cosine Similarity: 0.9261
GPT Descriptions:
Lawyer:A lawyer is a professional who is trained and licensed to practice law, representing clients in legal matters and providing advice on legal rights and obligations. They typically specialize in areas such as criminal law, civil rights, corporate law, or family law, and their work often involves drafting legal documents, negotiating settlements, and advocating in court. Distinguishing characteristics of lawyers include their adherence to ethical standards, the ability to interpret and apply laws, and strong skills in communication and critical thinking.
Attorney: An attorney is a legal professional who is qualified to represent clients in legal matters, provide legal advice, and advocate on their behalf in court. Typically, attorneys hold a law degree and are licensed to practice law in a specific jurisdiction. Distinguishing characteristics include their ability to engage in legal proceedings, draft legal documents, and navigate complex legal systems.

Interpretation.
Both words refer to the same licensed legal profession under different naming conventions. Their legal authority, job scope, and institutional roles are nearly indistinguishable, enabling the spy to imitate civilian descriptions almost perfectly without exposing unique semantic cues.

D.2 Medium Difficulty Examples

Example 4
Keyword Pair: (Knowledge, Wisdom)
Cosine Similarity: 0.7861
GPT Descriptions:
Knowledge: Knowledge refers to the understanding, awareness, or familiarity gained through experience or education. It encompasses facts, information, skills, and insights acquired over time, and is typically categorized into various types, such as theoretical, practical, or experiential. Distinguishing characteristics of knowledge include its ability to inform decision-making, enhance critical thinking, and contribute to personal and societal development.
Wisdom: Wisdom is the ability to make sound judgments and decisions based on knowledge, experience, and understanding. It often involves a deep insight into human nature and the complexities of life, allowing individuals to navigate challenges effectively. Unlike mere knowledge, which can be factual and theoretical, wisdom emphasizes the practical application of insights in real-world situations.

Interpretation.
The two concepts are strongly related but differ in abstraction and functional role. Knowledge emphasizes the possession of information and facts, while wisdom emphasizes the ability to apply knowledge through judgment and experience. This creates moderate semantic overlap with a clear but subtle conceptual boundary.

Example 5
Keyword Pair: (River, Lake)
Cosine Similarity: 0.7854
GPT Descriptions:
River: A river is a large, flowing body of freshwater that typically originates from sources such as springs or melted snow and moves toward a larger body of water, like an ocean or lake. It plays a crucial role in ecosystems, providing habitat for diverse wildlife and serving as a vital resource for human activities such as agriculture, transportation, and recreation. Rivers are distinguished by their continuous flow, defined banks, and their ability to shape landscapes through erosion and sediment deposition.
Lake: A lake is a large body of freshwater or saltwater that is surrounded by land. Typically, lakes are classified by their size, depth, and the presence of aquatic life, serving as habitats for various species and recreational areas for activities like fishing and boating. Unlike rivers, lakes are usually static, with water that does not flow in a defined direction.

Interpretation.
Both concepts refer to natural bodies of water and frequently co-occur in geographic and ecological contexts. However, a river is defined by dynamic flow, while a lake is defined by static containment. The shared environmental domain with distinct physical dynamics results in moderate semantic separability.

Example 6
Keyword Pair: (Bottle, Jug)
Cosine Similarity: 0.7820
GPT Descriptions:
Bottle: A bottle is a rigid container, typically made of glass or plastic, used for storing liquids. It usually features a narrow neck and often has a cap or cork to seal the contents. Bottles are commonly used for beverages, pharmaceuticals, and various household products, distinguishing themselves by their shape and ability to preserve the integrity of their contents.
Jug: A jug is a container typically used for holding and pouring liquids, often featuring a handle and a spout. Commonly made from materials like glass, ceramic, or plastic, jugs are used for serving beverages such as water, juice, or milk. They are distinguished by their larger capacity compared to standard cups or glasses, making them ideal for communal serving.

Interpretation. Both objects are liquid containers used in everyday life and often appear in similar usage contexts. However, they differ in shape, capacity, and functional design, with jugs emphasizing pouring and larger volume. This functional variation creates a medium level of semantic distinction.

D.3 Easy Difficulty Examples

Example 7
Keyword Pair: (Mountain, Peak)
Cosine Similarity: 0.7189
GPT Descriptions:
Mountain: A mountain is a large landform that rises prominently above its surroundings, typically characterized by steep slopes, a significant elevation, and a defined summit. Mountains are often formed through tectonic forces or volcanic activity and are commonly found in ranges. They serve various ecological, recreational, and cultural roles, providing habitats for diverse wildlife and opportunities for activities like hiking and skiing.
Peak: The word 'peak' refers to the highest point or summit of a structure, object, or phenomenon, such as a mountain or a performance level. It is commonly used in various contexts, including geography (to describe mountain tops), business (to indicate maximum performance or sales), and time (to denote the highest occurrence of an event). Distinguishing characteristics of 'peak' include its association with the concept of elevation and the idea of reaching a maximum or optimal state.

Interpretation.
A mountain refers to a complete large-scale landform, while a peak refers specifically to the highest point of such a landform. The concepts are hierarchically related rather than interchangeable, creating a clear semantic asymmetry and low identity ambiguity.

Example 8
Keyword Pair: (Piano, Drum)
Cosine Similarity: 0.7183
GPT Descriptions:
Piano: A piano is a large musical instrument with a keyboard that produces sound by striking strings with hammers when the keys are pressed. It is commonly used in various musical genres, including classical, jazz, and pop, and is known for its versatility and expressive range. The distinguishing characteristics of a piano include its 88 keys (in a standard model), a rich tonal quality, and the ability to play both melody and harmony simultaneously
Drum: A drum is a cylindrical percussion instrument that produces sound when its membrane, or drumhead, is struck with sticks or hands. Commonly used in various music genres, from marching bands to rock and jazz, drums are distinguished by their ability to create rhythmic patterns and dynamic beats. Drums can vary in size, shape, and material, affecting their tone and resonance.

Interpretation.
Both are musical instruments used in performance contexts, but they belong to entirely different instrument families. The piano produces pitched melodic and harmonic structures, while the drum produces unpitched rhythmic patterns. This functional and acoustic contrast leads to low semantic ambiguity.

Example 9
Keyword Pair: (Panda, Koala)
Cosine Similarity: 0.7148
GPT Descriptions:
Panda: A panda is a large, bear-like mammal native to China, known for its distinctive black and white fur and round face. Typically associated with the giant panda, this species primarily feeds on bamboo, which constitutes the majority of its diet, and is recognized as a symbol of wildlife conservation efforts. Pandas are generally solitary animals and have a playful demeanor, making them popular in zoos and as cultural icons.
Koala:A koala is a herbivorous marsupial native to Australia, known for its distinctive large ears, round face, and gray fur. Typically found in eucalyptus forests, koalas primarily feed on eucalyptus leaves, which are low in nutrients and require them to spend most of their time sleeping or resting. Unlike many other marsupials, koalas have a unique vocalization and a solitary lifestyle, making them a symbol of Australian wildlife.

Interpretation.
Both animals are iconic, bear-like mammals often perceived as visually similar by non-experts. However, they differ fundamentally in biological classification (placental mammal vs. marsupial), habitat, and diet. This surface similarity with deep biological differences yields low semantic overlap and clear identifiability.

E Examples of LLM-Generated Dynamic Cheatsheet Strategies

E.1 Single-Agent Cheatsheet Memory

The following examples are drawn from the **single-agent cheatsheet memory**, where strategies primarily focus on description quality, semantic consistency, and individual voting behavior:

- "Note players whose descriptions diverge from the common theme across multiple rounds."
- "Consistently vague or generic descriptions suggest a player may be the spy."
- "Monitor for players who maintain vague descriptions over multiple rounds as potential spies."
- "Observe voting patterns; consistent targeting of a player suggests they may be the spy."
- "Evaluate descriptions for specificity; generic descriptions are red flags in identifying the spy."

Compared to multi-agent settings, these strategies operate at an **individual behavioral level**, emphasizing how a single player can infer deception from linguistic vagueness, semantic drift, and local voting signals.

Interpretation.
These strategies demonstrate that the single-agent cheatsheet mainly captures *local behavioral heuristics*, such as tracking whether a player's descriptions increasingly diverge from the group consensus or remain persistently vague. This aligns closely with how human players individually detect deception based on linguistic uncertainty and micro-level inconsistencies.

E.2 Multi-Agent Cheatsheet Memory

The following examples are drawn from the **multi-agent cheatsheet memory**, where strategies emphasize collective voting dynamics and group-level behavioral patterns:

- "Consistently track and analyze unanimous voting patterns to identify potential spies."
- "Use unanimous or near-unanimous voting patterns against a player over multiple rounds as a strong indicator of their spy status."
- "Focus on players who are voted against consistently, even if not unanimously, as they may be the spy."
- "Streamline the game by quickly eliminating players who receive multiple rounds of high votes to narrow down the spy's identity."
- "Consider players who receive no or very few votes as potential nonspies, but remain vigilant for strategic voting patterns."

Compared to the single-agent case, these strategies operate at a **group behavioral level**, capturing emergent dynamics such as majority pressure, coordinated suspicion, and elimination acceleration.

Interpretation.
These examples verify that the Dynamic Cheatsheet does not store arbitrary conversation logs, but instead condenses cross-game experiences into **compact, interpretable, and transferable strategy rules**. The stored strategies align closely with human heuristics in social deduction games, such as tracking semantic drift, watching for persistent vagueness, and exploiting voting regularities. This qualitative evidence supports the claim that the proposed framework enables genuine *test-time strategic learning* rather than superficial memory replay.

F Implementation Details

For all models used in our experiments, we query them via their corresponding APIs. We access **GPT-4o-mini** through the **CMU AI Gateway**,[a] **Qwen2.5-72B-Instruct** and **DeepSeek-V3** through the **SiliconFlow platform**,[b] **Llama 3.1 70B Instruct Turbo** through the **Together AI platform**,[c] and **Gemini-2.5-Flash-Lite** through **Google AI Studio**.[d]

As for **Dynamic Cheatsheet**, this framework is implemented as a **training-free external memory module** integrated directly into each agent's prompting loop. Each agent is equipped with a `SpyCheatSheetManager`, which maintains a bounded memory pool of at most **10 strategy entries** and supports **vector-based retrieval**. All strategy texts are embedded using the `BAAI/bge-m3` embedding model served via the **SiliconFlow API**. To reduce **retrieval cost** and **storage overhead**, we apply **PCA-based dimensionality reduction** to compress embeddings into **128 dimensions** before indexing.

All strategy embeddings are indexed using **FAISS** with **inner-product similarity search**. The FAISS index, compressed PCA matrix, and raw strategy texts are automatically serialized to disk and reloaded across different runs to support **persistent cross-game learning**. Strategy synthesis is performed by a dedicated **curator agent** after every $K = 5$ games, which takes as input the retrieved past strategies and the full public game log, and outputs new reusable **one-line strategy rules**. All retrieval, memory updates, and curator synthesis are conducted at **inference time** without any model finetuning.

To ensure **reproducibility**, we fix the **random seed to 42** for all experiments.

[a] https://ai-gateway.andrew.cmu.edu/
[b] https://siliconflow.cn/
[c] https://www.together.ai/
[d] https://aistudio.google.com/