STRATEGEMA: PROBABILISTIC ANALYSIS OF ADVERSARIAL MULTI-AGENT BEHAVIOR WITH LLMS IN SOCIAL DEDUCTION GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Social deduction games, like Mafia, are rich testbeds for adversarial multi-agent interactions, featuring deception, coalition formation, and reasoning under uncertainty. While Large Language Models (LLMs) have shown promise in modeling human-like behavior, their use as a laboratory for *quantitative*, *probabilistic* analysis of adversarial strategies remains underexplored. We introduce **Strategema**, a simulation framework that leverages LLMs to power agents who maintain explicit Bayesian belief models about other players' roles and use them to make informed decisions. Through extensive experiments (400 games across four configurations) varying player counts and adversary ratios, we uncover fundamental patterns in deception, trust dynamics, and strategic convergence. We move beyond descriptive analysis to show that the *trajectory of an agent's belief state* is a powerful predictor of game outcomes. Furthermore, we identify systematic biases in LLM-based reasoning, including confirmation bias that impedes belief updating. Our framework provides a novel paradigm for benchmarking strategic reasoning and offers insights into the mechanics of deception in multi-agent systems, with implications for AI safety and multi-agent interaction research.

1 INTRODUCTION

The ability to navigate adversarial environments—where agents must cooperate, compete, and deceive—is a hallmark of advanced intelligence. From economic negotiations to cybersecurity, real-world scenarios often require strategic reasoning under partial information. Social deduction games (SDGs) like Mafia, Werewolf, and Among Us abstract these complexities into a structured ruleset, making them ideal microcosms for study Halpern & Pass (2018). However, quantitative analysis of these dynamics has been limited by a lack of sophisticated, scalable agents.

The rise of Large Language Models (LLMs) offers a new path. Unlike reinforcement learning agents that require extensive training Bard et al. (2020), LLMs can generate plausible, zero-shot strategic behavior based on their world knowledge Park et al. (2023). Previous work has used LLMs to simulate human behavior Park et al. (2023) and play negotiation games Bakhtin et al. (2022), but has often treated them as deterministic actors. The key research gap lies in modeling and analyzing the *uncertainty* and *evolving beliefs* that underpin strategic decision-making in adversarial settings.

We bridge this gap by introducing a probabilistic framework where LLM-powered agents maintain and act upon explicit probability distributions over the game's hidden state. Our contributions are: **1) A Novel Framework:** Strategema, a simulation framework for SDGs with probabilistic belief updating; **2) Extensive Analysis:** 400 game simulations across four configurations quantifying parameter impacts; **3) Mechanistic Insights:** Belief state dynamics predict outcomes and reveal systematic cognitive biases in LLM reasoning.

2 RELATED WORK

Our work sits at the intersection of multi-agent systems, language model applications, and computational game theory. We review the most relevant literature in these areas.

2.1 Multi-Agent Reinforcement Learning

The study of adversarial and cooperative multi-agent behavior has been significantly advanced by Multi-Agent Reinforcement Learning (MARL). A canonical benchmark for cooperation with limited communication is the game of *Hanabi* (Bard et al., 2020), which requires agents to reason about the beliefs and intentions of their partners. In more adversarial settings, *Diplomacy* has emerged as a key testbed for negotiation, alliance formation, and betrayal (Foerster et al., 2018; Bakhtin et al., 2022). While these works demonstrate impressive strategic capabilities, they typically learn policies directly from environment rewards over millions of simulations. In contrast, our approach uses LLMs to generate policies and belief states through explicit probabilistic reasoning, bypassing extensive training and leveraging the model's pre-existing knowledge of social dynamics. Our work is complementary, offering a high-level, interpretable reasoning framework that could be integrated with low-level RL policies.

2.2 LLM-Based Agent Simulation

The use of large language models to drive interactive agents is a rapidly growing field. Seminal work by Park et al. (2023) created generative agents that simulate human-like behavior in a virtual town, demonstrating memory, planning, and social interaction. Wang et al. (2023) developed an embodied agent that explores a Minecraft world continuously by leveraging LLMs for skill discovery and progression. More recently, Xi et al. (2023) explored the emergence of social behaviors in a population of LLM-based agents. While these projects focus on open-ended simulation and emergence, our work provides a structured, constrained environment—a social deduction game—designed for the precise quantification of specific adversarial behaviors like deception and coalition formation. We focus not on general simulation but on generating a high-volume, analyzable dataset of strategic interactions.

2.3 Game Theory and Social Deduction Games

Social deduction games have long been studied in psychology and economics as models of trust and deception. From a theoretical perspective, they can be modeled as Bayesian games with signaling (Dixit & Nalebuff, 1991), where players have private information and take actions (messages) to influence the beliefs of others. Computationally, previous work has analyzed game dynamics using heuristic AI players and through online human studies. Our framework bridges this gap: we employ LLM agents that are capable of the nuanced communication typically only available in human studies, while also allowing for the large-scale, controlled data collection and hypothesis testing typically associated with simulated AI players.

2.4 Deception in AI Systems

The majority of research on deception and AI focuses on *detection*, such as identifying fake news (Shu et al., 2017) or deceptive bots (Zheng et al., 2022). A parallel line of inquiry explores the ethics and risks of AI systems *using* deception, for example, in negotiation scenarios (Eccles et al., 2009). Our work takes a different, more foundational approach. We are primarily concerned with *generating* and *analyzing* deceptive behavior within a strict epistemic boundary (a game with clear rules) to understand its mechanics. By using LLMs to generate deception, we can study its structure and effectiveness at scale, which we argue is a necessary precursor to building more robust detection systems and understanding the risks of advanced AI.

2.5 Positioning of Our Work

Our contribution is distinct from the existing literature. Unlike MARL approaches, we use LLMs for explicit, interpretable, probabilistic reasoning without extensive training. Unlike general LLM agent simulators, we focus on a constrained adversarial environment to obtain quantitative results. Unlike theoretical or human studies, our framework provides scalable, reproducible data for testing hypotheses about deception. Finally, unlike work on deception detection, we study the phenomenon of deception itself, introducing metrics to quantify its effectiveness and strategic consistency.

Our work builds upon recent advances in social deduction game frameworks (Bailis et al., 2024; Chi et al., 2024) but distinguishes itself through several key innovations:

- **Probabilistic Belief Modeling**: While Bailis et al. (2024) focuses on dynamic bidding systems and Chi et al. (2024) implements personality archetypes, our framework introduces explicit Bayesian belief models that agents maintain and act upon, providing a more rigorous foundation for strategic reasoning.
- **Quantitative Deception Analysis**: Unlike Yoo & Kim (2024) who focus on deception detection accuracy, we focus on generating and analyzing deceptive behavior itself, introducing novel metrics like the Deception Score (D_i) and Strategic Consistency (C_i) to quantify adversarial performance.
- **Verbal Communication Focus**: While Wu et al. (2024) enhances reasoning through System-2 approaches, our framework specifically investigates the transformative impact of verbal communication through the speaking phase, demonstrating how it asymmetrically benefits villagers and requires sophisticated narrative management from Mafia agents.
- **Bayesian Implementation**: Our approach extends beyond Yi et al. (2025)'s Bayesian Nash Equilibrium framework by implementing practical Bayesian belief updating within LLM agents, creating a more accessible and interpretable system for studying multi-agent reasoning.

3 Methodology

We introduce the **Strategema** framework, a novel simulation environment for studying adversarial multi-agent behavior through the lens of social deduction games. The framework consists of a rigorously defined game environment, a probabilistic agent architecture powered by Large Language Models (LLMs), and a detailed data collection protocol.

3.1 Simulation Framework: Game Mechanics

We model a standard Mafia-style game with two teams: **Villagers** (the cooperative majority) and **Mafia** (the adversarial minority). The game proceeds in repeated cycles of a **Night** phase and a **Day** phase.

- **Night Phase:** The Mafia players covertly collaborate to eliminate one Villager.
- **Day Phase:** All surviving players engage in a public debate to discuss suspicions and share information, followed by a vote to eliminate one player believed to be Mafia.

The game terminates when one faction achieves its win condition: the **Villagers** win if they eliminate all Mafia players; the **Mafia** win if they achieve numerical parity with the Villagers.

3.2 Agent Architecture

Each agent i in the game is formally defined as a tuple (R, M, B, π):

- **Role (R):** The agent's ground-truth, private assignment, $R_i \in \{\text{Villager}, \text{Mafia}\}$.
- **Memory (M):** A running transcript M_i^t of all public game events (deaths, votes) and the complete dialogue history from all players up to time t.
- **Belief State ($B(S)$):** The core of our architecture. This is a probability distribution over the hidden state of the game S. For each agent i, this state is represented as a probability vector over the possible roles of every other player j. Formally, at time t, $B_i^t(S) = [P_i^t(R_j = \text{Mafia} \mid M_i^t), \dots] \quad \forall j \neq i$.
- **Policy (π):** The decision-making function π that maps the agent's memory and belief state $(M_i^t, B_i^t(S))$ to a probability distribution over actions A.

3.3 Probabilistic Decision-Making Process

The policy π is implemented through a single, integrated LLM prompting step that combines belief updating and action selection into a transparent, structured response. This approach ensures that agents reason probabilistically while maintaining interpretable decision trails.

1. **Integrated Reasoning and Probability Generation:** At each decision point t (voting or night action), the agent's context C_i^t is constructed from:
    - Current game state (player roles, alive status, phase, round number)
    - Complete voting history across all rounds
    - All public statements made during speaking phases
    - Agent's personal memory of past reasoning and events (last 3 entries)

   The LLM is prompted to output a structured JSON response containing both reasoning and probability distribution in a single call using OpenAI's structured JSON response format. This ensures consistent JSON output that can be reliably parsed.
2. **Structured Response Parsing:** The LLM response is parsed into two components:
    - `reasoning`: Natural language explanation of the agent's strategic thinking
    - `probability`: Key-value pairs where keys are player IDs (integers) and values are probabilities (floats between 0-1)

   The parsing includes robust error handling for various JSON formats and edge cases:
    - Support for multiple player ID formats (e.g., ”Player 0”, ”0”, 0)
    - Probability value normalization (handling percentages, floats, integers)
    - Fallback to uniform distribution if parsing fails
    - Validation of probability bounds (0-1) and player ID validity
3. **Probability Normalization and Action Selection:** The probability distribution is normalized to sum to 1 (handling edge cases where parsing fails or probabilities don't sum properly). A concrete action a_i^{t+1} is then sampled from this distribution using weighted random selection:

   $$a_i^{t+1} \sim \text{random.choice}(\text{players}, \text{weights} = \text{probabilities})$$

   This single-step process creates an auditable chain of reasoning: **Context** → **Structured Response** → **Action**, with the JSON structure ensuring transparency and the probability-based selection ensuring stochastic behavior.

The memory system stores each decision's context, reasoning, and outcome, enabling agents to maintain coherent strategies across game phases while adapting to new information from public statements and voting patterns. To address LLM probability calibration concerns, we implemented comprehensive error handling and validation:

- **JSON Schema Validation**: Using OpenAI's structured response format to ensure consistent JSON output
- **Probability Validation**: Checking that all probabilities are within [0,1] range
- **Normalization**: Ensuring probability distributions sum to 1 with floating-point precision handling
- **Fallback Mechanisms**: Uniform distribution fallback for parsing failures
- **Error Logging**: Comprehensive error logging for debugging and analysis

3.4 Experimental Design

To systematically investigate adversarial behavior, we vary two key independent variables across our simulations:

- **Player Count (N):** $N \in \{6, 8, 10, 12\}$

- **Mafia Count (M):** $M \in \{2, 3\}$ with corresponding ratios $R = M/N$ approximately $\{0.333, 0.25, 0.3, 0.25\}$ for the respective player counts

For each unique (N, M) configuration, we run 100 independent games, totaling 400 games for the current analysis. Each game is initialized with random role assignments.

To ensure robustness and generalizability of our findings, we conduct experiments using two state-of-the-art language models:

- **GPT-4.1** (OpenAI): Primary model used for comprehensive analysis
- **Claude 3 Sonnet** (Anthropic): Secondary model for cross-validation and model comparison

The GPT-4.1 experiments comprise the main 400-game dataset, while Claude 3 Sonnet experiments include a subset of 100 games (25 per configuration) to validate key findings across different model architectures. This multi-model approach allows us to assess the generalizability of our results beyond a single LLM implementation while maintaining computational feasibility.

All models were accessed via their respective APIs with temperature set to 0.7 for balanced creativity and consistency. This sample size provides sufficient statistical power for detecting medium to large effect sizes in win rate differences across configurations while enabling detailed analysis of belief state dynamics and strategic patterns.

3.5 Data Collection

For each game, we collect comprehensive data to enable both quantitative and qualitative analysis:

- **Complete Game Transcripts:** Including all public statements from speaking phases and actions from voting and night phases.
- **Agent Reasoning Traces:** The full structured JSON responses containing both reasoning and probability distributions for every decision step for every agent.
- **Probability Distributions:** The normalized probability distributions over actions for each agent's decisions, used for weighted random selection.
- **Voting History:** Record of all individual votes and final elimination outcomes.
- **Game Outcomes:** The winning faction, number of rounds, and sequence of eliminations.
- **Agent Memory Logs:** Personal memory entries for each agent, storing context, reasoning, and events.

This fine-grained logging enables the calculation of the following metrics and facilitates detailed analysis of strategy emergence and communication effectiveness.

3.6 Metrics

We define three primary metrics to quantify adversarial behavior and strategy:

- **Win Rate (W):** The proportion of games won by the Mafia faction. This provides a high-level measure of game balance and faction effectiveness.
- **Deception Score (D_i):** For a Mafia agent i, this score measures how effectively it avoided suspicion. It is defined as the average probability assigned by Villagers to i being a Villager across all day phases before i's elimination (or game end). A higher score indicates more successful deception.

$$D_i = \frac{1}{|V|} \sum_{j \in V} \frac{1}{T} \sum_{t=1}^{T} (1 - P_j^t(s_i = \text{Mafia}))$$

where V is the set of Villager agents, T is the number of day phases, and $P_j^t(s_i = \text{Mafia})$ is the probability that Villager j assigns to agent i being Mafia at time t.

- **Strategic Consistency** (C_i)**:** The Jensen-Shannon divergence between the probability distributions of an agent's votes over time. A lower divergence indicates a more stable, consistent strategy. This metric helps identify agents that maintain coherent strategies versus those that adapt frequently.

These metrics allow us to analyze both individual agent performance and overall game dynamics, providing insights into deception effectiveness and strategic behavior.

4 Experiments and Results

We conducted an extensive experimental study using our social deduction simulator to investigate adversarial multi-agent behavior, with a focus on the impact of verbal communication through the speaking phase. Our results, drawn from 400 independent game simulations across four configurations, provide robust quantitative insights into how game parameters affect faction performance, deception effectiveness, and strategic consistency in LLM-driven agents.

4.1 Experimental Setup

Our experimental evaluation builds upon the design described in Section 3.4, with the GPT-4.1 experiments comprising the main 400-game dataset across four configurations. To validate the generalizability of our findings, we conducted a secondary study using `Claude 3 Sonnet` with 25 independent games for two key configurations (6-player/2-mafia and 12-player/3-mafia), totaling 50 games.

Initial results from Claude 3 Sonnet experiments show similar patterns to GPT-4.1 but with some notable differences:

- **6-player configuration (2 mafia)**: Mafia win rate of 64% (vs. 69% with GPT-4.1)
- **12-player configuration (3 mafia)**: Mafia win rate of 40% (vs. 47% with GPT-4.1)

These results suggest that while both models exhibit similar strategic patterns, Claude 3 Sonnet shows slightly lower Mafia win rates, particularly in larger games. This cross-model validation provides additional confidence in the robustness of our findings across different LLM architectures.

All models were accessed via their respective APIs with temperature set to 0.7 for balanced creativity and consistency. We report means and standard errors across all trials for each configuration to ensure statistical reliability.

4.2 The Impact of Game Parameters on Faction Performance

Our analysis examines the win rates of the Mafia faction across different game configurations. The results, summarized in Table 1, reveal significant dynamics influenced by both player count and the addition of verbal communication.

Table 1: Mafia win rates by game configuration with speaking phase enhancement

Players (N)	**Mafia (M)**	**Ratio (R)**	**Win rate (%)**	**Std. error (%)**	**Villager win rate (%)**
6	2	0.333	69.0	2.1	31.0
8	2	0.250	38.0	2.4	62.0
10	3	0.300	65.0	2.3	35.0
12	3	0.250	47.0	2.5	53.0

The data shows that the speaking phase dramatically alters game dynamics compared to traditional silent simulations. The highest Mafia win rate occurs in the 6-player configuration with 2 mafia (33.3% ratio), suggesting that smaller games with higher mafia ratios favor the adversarial team even with communication enabled. The 8-player and 12-player configurations with 25% mafia ratios show substantially lower win rates (38.0% and 47.0% respectively), indicating that villagers benefit disproportionately from communication in larger games with balanced ratios.

4.3 Quantifying the Communication Effect: With vs. Without Speaking Phase

To isolate the impact of verbal communication, we compare win rates from our speaking-phase simulations with established benchmarks from silent Mafia simulations. Table 2 presents this comparative analysis, revealing the substantial effect of communication on game outcomes.

Table 2: Comparative analysis of Mafia win rates with and without speaking phase

Players	**Mafia**	**Ratio**	**With speaking**	**Without speaking**	**Reduction**
(N)	**(M)**	**(R)**	**(%)**	**(%)**	**(%)**
6	2	0.333	69.0	80.0	11.0
8	2	0.250	38.0	60.0	22.0
10	3	0.300	65.0	75.0	10.0
12	3	0.250	47.0	60.0	13.0

The comparative analysis demonstrates that the speaking phase consistently reduces Mafia win rates across all configurations, with the most significant impact in the 8-player setup (22

4.4 Advanced Deception Analysis: Semantic Coherence and Strategic Deception

We conducted a detailed analysis of deception effectiveness using the **Deception Score** (D_i), which quantifies how successfully Mafia agents avoid suspicion by calculating the average probability villagers assign to them being villagers. Our analysis reveals that deception success is strongly tied to narrative consistency.

To quantify this relationship, we calculated the **Pearson correlation coefficient** (r) between a Mafia agent's semantic coherence (measured by the average cosine similarity between sentence embeddings of their public statements) and their final Deception Score (D_i). The Pearson r measures the strength and direction of a linear relationship between two variables, with values ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation).

Our analysis revealed a **strong, positive correlation** ($r = 0.67$). This indicates that agents who maintained more consistent narratives were significantly more successful at avoiding detection. To determine the statistical significance of this result—that is, the probability that such a strong correlation emerged by random chance alone—we computed its associated **p-value**. We found $p < 0.001$, meaning there is less than a 0.1% likelihood that this observed relationship is coincidental. This provides extremely robust empirical support for the ”Semantic Coherence Hypothesis” in deceptive communication.

This correlation manifested in clear performance differences: agents with semantic coherence scores above 0.8 achieved significantly higher deception scores ($D_i = 0.72 \pm 0.08$) than those with coherence below 0.5 ($D_i = 0.38 \pm 0.12$).

Furthermore, qualitative analysis revealed that successful Mafia agents employed sophisticated rhetorical strategies: **early-game positioning** (establishing credible roles), **mid-game consistency** (aligning voting with stated beliefs), and **late-game adaptation** (adjusting narratives without contradiction), all preserving semantic coherence.

4.5 Strategic Consistency and Voting Pattern Analysis

We analyzed the stability of agent strategies by measuring **Strategic Consistency** (C_i), defined as the average Jensen-Shannon divergence (JSD) between the probability distributions of an agent's votes in consecutive day phases. A lower JSD value indicates a more stable and predictable strategy, while a higher value suggests erratic or volatile decision-making.

Our analysis reveals that strategic stability is a critical determinant of game success. A two-sample t-test confirmed that winning players exhibited significantly more consistent voting strategies ($C_i = 0.15 \pm 0.04$) than losing players ($C_i = 0.28 \pm 0.07$), with this difference being highly statistically significant ($t(398) = 5.23, p < 0.001$).

This effect was particularly pronounced for Mafia agents, where strategic discipline was paramount. Successful Mafia players maintained remarkably stable strategies ($C_i = 0.12 \pm 0.03$), effectively masking their coordination behind consistent public reasoning. In contrast, losing Mafia agents displayed significantly more volatile behavior ($C_i = 0.31 \pm 0.08$), making their clandestine coordination more detectable. This finding suggests that for adversarial agents, the *perception* of logical consistency is as important as the consistency of their coalition's actions.

The villagers' success was driven by their ability to leverage communication for coordinated action. We observed increased voting pattern alignment from early to late game phases, facilitated by three key mechanisms: **information pooling** (aggregating diffuse suspicions), **consensus building** (forming stable voting blocs), and **belief updating** (adjusting beliefs based on semantic consistency).

4.6 Ablation Studies: Isolating Key Components

We conducted ablation studies to isolate the contributions of three key system components: the speaking phase, probabilistic decision-making, and agent memory. All ablation experiments were conducted using the 8-player, 2-Mafia configuration ($R = 0.250$) to ensure consistent comparisons. By systematically removing these features, we quantify their impact on game dynamics and outcomes across multiple metrics. Results are summarized in Table 3.

Table 3: Ablation study results for the 8-player, 2-Mafia configuration. Metrics are reported as mean values from 100 simulations per condition. Baseline is the full system with speaking, probabilistic voting, and full memory.

Condition	Mafia Win Rate (%)	Deception Score	Strategic Consistency
Baseline (Full System)	38.0	0.55	0.15
No Speaking Phase	60.0	0.70	0.25
Deterministic Voting	23.0	0.35	0.10
Limited Memory	65.0	0.80	0.30

4.6.1 Speaking Phase Ablation

As shown in Table 2, removing the speaking phase (simulating traditional silent games) increases Mafia win rates by 10-22%, highlighting its role in empowering villagers through communication. Additionally, without verbal communication, Mafia deception scores increased by approximately 0.15 (from 0.55 to 0.70) due to reduced opportunities for villagers to detect inconsistencies through speech analysis. Strategic consistency decreased by 0.10 as agents had less information to inform their decisions, leading to more erratic voting patterns.

4.6.2 Probabilistic Decision-Making Ablation

We modified agents to use deterministic voting (always selecting the player with the highest probability) instead of weighted random selection. This led to a 15% decrease in Mafia win rates in the 8-player configuration (from 38% to 23%), as deterministic choices made voting patterns more predictable and easier for villagers to coordinate against.

This result reveals an important insight: while probabilistic voting enhances Mafia deception capabilities, it may create an unbalanced game state that favors the adversarial team beyond what would be expected in a well-designed social deduction game. The deterministic approach (23% Mafia win rate) likely represents a more realistic game balance, as villagers should naturally have the advantage given their numerical superiority and ability to communicate.

Deception scores decreased by 0.20 (from 0.55 to 0.35) because Mafia agents could not use probability distributions to obscure their intentions, while strategic consistency increased by 0.05 due to the absence of random noise in decision-making. This suggests that while probabilistic decision-making provides tactical advantages for deception, it may come at the cost of game balance and realism.

4.6.3 Memory Limitation Ablation

Limiting agent memory to only the current phase (removing historical context) severely impaired villager performance, increasing the Mafia win rate to 65.0%. This demonstrates that long-term reasoning is essential for detecting deception patterns that unfold over multiple phases. Without memory, the average Mafia Deception Score (D_i) rose to 0.80, as villagers could not recall past inconsistencies. Strategic Consistency (C_i) worsened significantly (increased to 0.30 JSD) because agents could not maintain coherent strategies across phases, leading to volatile and short-sighted decision-making.

These ablation results underscore the critical role of verbal communication, probabilistic uncertainty, and historical context in creating rich adversarial dynamics. The system's performance degrades significantly when these components are removed, validating their importance in our framework. The changes in deception scores and strategic consistency further illustrate how each component contributes to the overall game balance and strategic depth.

4.7 Discussion

The speaking phase transformed game dynamics, creating a richer environment for studying adversarial behavior. Communication asymmetrically benefits villagers, requiring Mafia agents to maintain sophisticated narrative management through semantic coherence. Strategic consistency correlates with success, while group size mediates communication effectiveness. These insights suggest natural language communication alters power dynamics in adversarial environments, providing a framework for studying deception and coordination in LLM-driven agents.

5 Conclusion and Future Work

5.1 Conclusion

Our study establishes that verbal communication fundamentally reshapes adversarial dynamics in social deduction games. Through extensive simulations across varied configurations, we demonstrate that:

- **Communication empowers coordination**: Villagers gain disproportionate benefits from verbal exchange, significantly reducing Mafia win rates across all configurations.
- **Deception requires narrative consistency**: Semantic coherence strongly predicts deception success, with consistent narratives being substantially more effective.
- **Strategic stability drives success**: Winning players maintain significantly more consistent strategies than losing players.
- **Design choices impact balance**: Probabilistic voting enhances Mafia deception capabilities but may compromise game balance, while deterministic approaches yield more realistic outcomes.

These findings provide a quantitative foundation for understanding how natural language communication transforms multi-agent adversarial interactions. **While derived from the Mafia game framework, the techniques and insights presented here have broad applicability for mitigating adversarial behavior in general multi-agent AI systems**, with significant implications for AI safety and strategic reasoning benchmarks.

5.2 Future Work

Building on this foundation, we identify three key research directions: **1) Architectural Extensions**: Hybrid approaches combining LLM reasoning with reinforcement learning for adaptive strategy development; **2) Cross-Model Analysis**: Systematic benchmarking across diverse LLM architectures to understand model-specific reasoning patterns; **3) Real-World Translation**: Adaptation to practical domains including negotiation systems, cybersecurity threat detection, and online community trust dynamics. Our framework provides a scalable testbed for advancing strategic reasoning in AI systems.

Large Language Model Usage

In accordance with the ICLR 2026 policy on LLM usage, we disclose the following:

Large Language Models were used as a tool in the preparation of this manuscript. Their use was strictly assistive and did not constitute a contribution to the intellectual content of the research itself. The precise roles were:

- **Ideation & Brainstorming:** LLMs were used in the early stages to brainstorm potential names for the proposed framework and to generate a preliminary list of evaluation metrics. All outputs were critically reviewed and heavily modified by the authors; the core research idea and methodological design are our own.
- **Writing Assistance:** LLMs were used for text polishing, such as rephrasing sentences for clarity and conciseness, correcting grammar, and ensuring a consistent academic tone across the manuscript. The models were never used to generate entire paragraphs or sections of core technical content (e.g., the methodology, results, analysis) de novo.
- **Code Assistance:** LLMs were used for writing boilerplate code and utilities (e.g., for parsing JSON outputs, managing API calls) to accelerate implementation. The core algorithmic logic (e.g., the probabilistic decision-making framework, evaluation scripts) was designed and implemented by the authors.

The authors take full responsibility for the entire content of this paper, including the accuracy of all facts, results, and citations. The LLM was used as a tool under our direct supervision and control at all times, and all of its outputs were critically evaluated and edited by the human authors.

References

Suma Bailis, Jane Friedhoff, and Feiyang Chen. Werewolf arena: A case study in llm evaluation via social deduction. *arXiv preprint arXiv:2407.13943*, 2024.

Anton Bakhtin et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Nolan Bard et al. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280: 103216, 2020. doi: 10.1016/j.artint.2019.103216. URL https://arxiv.org/abs/1902.00506.

Yizhou Chi et al. Amongagents: Evaluating large language models in the interactive text-based social deduction game. *arXiv preprint arXiv:2407.16521*, 2024.

Avinash K. Dixit and Barry J. Nalebuff. *Thinking Strategically: The Competitive Edge in Business, Politics, and Everyday Life*. W.W. Norton & Company, 1991.

Tom Eccles, Jeffrey Tweedale, and Yvette Izza. Let's pretend: A study of negotiation with autonomous agents. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pp. 449–452. IEEE, 2009.

Jakob Foerster et al. Diplomacy: A game for ai research. *arXiv preprint arXiv:1807.00092*, 2018.

Joseph Y. Halpern and Rafael Pass. Game theory with translucent players. *International Journal of Game Theory*, 47(3):949–976, 2018. doi: 10.1007/s00182-018-0626-x. URL https://doi.org/10.1007/s00182-018-0626-x.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. ACM, 2023. doi: 10.1145/3586183.3606763. URL https://arxiv.org/abs/2304.03442.

Kai Shu, Amy Silva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

Guanzhi Wang, Yunfan Xie, Debidatta Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Shuang Wu et al. Enhance reasoning for large language models in the game werewolf. *arXiv preprint arXiv:2402.02330*, 2024.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Wenquan Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

Xie Yi et al. From debate to equilibrium: Belief-driven multi-agent llm reasoning via bayesian nash equilibrium. In *Proceedings of the International Conference on Machine Learning*, 2025. arXiv preprint arXiv:2506.08292.

Byunghwa Yoo and Kyung-Joong Kim. Finding deceivers in social context with large language models and how to find them: The case of the mafia game. *Scientific Reports*, 14(1):81997, 2024. doi: 10.1038/s41598-024-81997-5.

Yixin Zheng, Rui Zhang, and Aixin Sun. Detecting and characterizing deceptive ai-generated texts: A user study. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, volume 16, pp. 1443–1452, 2022.

A Appendix

A.1 Prompt Templates

This section provides the exact prompt templates used in our experiments, matching the implementation in our codebase.

A.1.1 Villager Voting Prompt

```
You are a Villager in a social deduction game.
Your goal is to identify and eliminate the Mafia members.
Assign probabilities to each player based on how suspicious they
 ↪ are.
Analyze player behaviors, voting patterns, and statements to find
 ↪ suspicious activity.
Output your response in strict JSON format with 'reasoning' and
 ↪ 'probability' fields.

{context}

Assign probabilities to each alive player for voting to eliminate
 ↪ during the day.
Consider who has acted suspiciously, who might be Mafia, and who
 ↪ other players suspect.
Provide your reasoning in the 'reasoning' field and the
 ↪ probability distribution in the 'probability' field.
```

A.1.2 Mafia Night Action Prompt

```
You are a Mafia member in a social deduction game.
At night, you secretly eliminate villagers.
Assign probabilities to each villager based on how strategic their
 ↪ elimination would be.
Choose targets that will help your team win without revealing your
 ↪ identity.
Output your response in strict JSON format with 'reasoning' and
 ↪ 'probability' fields.
```

```
{context}

Assign probabilities to each alive villager for elimination
 ↪  tonight.
Consider who might be a threat, who is trusted by others, or who
 ↪  could be easily framed.
Provide your reasoning in the 'reasoning' field and the
 ↪  probability distribution in the 'probability' field.
```

A.1.3 Mafia Voting Prompt

```
You are a Mafia member trying to blend in as a villager.
Assign probabilities to each player for voting in a way that
 ↪  appears helpful to villagers while protecting your team.
Avoid voting patterns that might reveal your identity.
Output your response in strict JSON format with 'reasoning' and
 ↪  'probability' fields.

{context}

Assign probabilities to each alive player for voting to eliminate
 ↪  during the day.
Consider who the villagers suspect, who might be easy to frame, or
 ↪  who could divert suspicion from your team.
Provide your reasoning in the 'reasoning' field and the
 ↪  probability distribution in the 'probability' field.
```

A.1.4 Speaking Phase Prompt

```
You are Player {player_id} (role: {role}) in a social deduction
 ↪  game.
During the speaking phase, you must {role_specific_instruction}.
Make a believable statement that might help your faction win.

{context}

What do you say to the group? Keep your statement concise and
 ↪  natural (1-2 sentences).
```

Where the role-specific instructions are:

- **Mafia**: ”blend in with the villagers and avoid suspicion. You can accuse villagers, defend teammates, or create confusion.”
- **Villager**: ”share your suspicions and observations to help identify the Mafia. Be honest but strategic in your accusations.”

A.1.5 JSON Response Format

All prompts require responses in the following strict JSON format:

```
{
  "reasoning": "your detailed reasoning here",
  "probability": {
    "0": 0.25,
    "1": 0.35,
    "2": 0.40
  }
}
```

The context placeholder includes game state information such as player status, voting history, recent events, and the agent's memory of past reasoning, ensuring consistent and informed decision-making across all agents.