From Belief Entrenchment to Robust Reasoning in LLM Agents

Jihwan Oh[1]* **Minchan Jeong**[1]* **Jongwoo Ko**[2]† **Se-Young Yun** [1]†
[1]KAIST AI [2]Microsoft
{ericoh929, mcjeong}@kaist.ac.kr

Abstract

Multi-Agent Debate (MAD) has emerged as a promising inference scaling method for Large Language Model (LLM) reasoning. However, it frequently suffers from belief entrenchment, where agents reinforce shared errors rather than correcting them. Going beyond merely identifying this failure, we decompose it into two distinct root causes: (1) the model's biased *static initial belief* and (2) *homogenized debate dynamics* that amplify the majority view regardless of correctness. To address these sequentially, we propose **DReaMAD** (**D**iverse **Rea**soning via **M**ulti-**A**gent **D**ebate). Our framework first rectifies the static belief via strategic prior knowledge elicitation, then reshapes the debate dynamics by enforcing perspective diversity. Validated on our new *MetaNIM Arena* benchmark, **DReaMAD** significantly mitigates entrenchment, achieving a +9.5% accuracy gain over ReAct prompting and a +19.0% higher win rate than standard MAD.

1 Introduction

While Large Language Models (LLMs) have shown impressive problem-solving capabilities (Achiam et al., 2023; Dubey et al., 2024), their ability to self-correct remains inconsistent. Single-agent mechanisms like Self-Refinement (Madaan et al., 2024) or Self-Consistency (Wang et al., 2022) often degrade performance when models cannot accurately assess their own reasoning (Huang et al., 2024). To address this, Multi-Agent Debate (MAD; Chan et al. 2023; Du et al. 2023; Liang et al. 2023) has emerged as a promising paradigm for inference-time scaling, leveraging inter-agent critique to refine outputs.

However, we observe that standard MAD is brittle in sequential, strategic environments, where a single suboptimal decision triggers consecutive failures that the consensus process fails to correct. Rather than challenging errors, homogeneous agents frequently converge into an "echo chamber," amplifying inherent cognitive biases. As we demonstrate in Section 3, this failure stems from a debate dynamic that prioritizes initial response frequency over logical validity, effectively rewarding popularity rather than correctness. Unlike static tasks (Cobbe et al., 2021; Edwards, 1994; He et al., 2020) where errors can be masked, strategic games enforce a rigorous penalty for every suboptimal reasoning step, making them ideal for dissecting cascading failures.

To address these limitations, we introduce *MetaNIM Arena*, a rigorous benchmark for adversarial strategic decision-making. Using this environment, we identify a critical failure mode termed **belief entrenchment**, where interaction actually hinders error correction. We decompose this phenomenon into two root causes:

(1) Static Initial Belief: The model's inherent bias prior to interaction, where it relies on immediate context rather than strategic foresight.

(2) Homogenized Debate Dynamics: A self-reinforcing mechanism where the persuasiveness of an argument is dictated by its initial popularity among homogeneous agents, suppressing valid but less probable reasoning paths.

Building on the *Learning from Multiple Approaches* framework (Council et al., 2005), which suggests that engaging with multiple problem-solving representations mitigates bias, we propose **DReaMAD** (**D**iverse **Rea**soning via **M**ulti-**A**gent **D**ebate with Refined Prompt). As shown in Figure 1, **DReaMAD** addresses the identified root causes by (1) eliciting strategic prior knowledge to correct static belief and (2) systematically modifying

*Equal contribution. †Corresponding authors.

Transactions of the Association for Computational Linguistics, vol. 14, pp. 1286–1307, 2026. https://doi.org/10.1162/TACL.a.728
Action Editor: Antoine Bosselut. Submission batch: 10/2025; Revision batch: 12/2025; Published 7/2026.

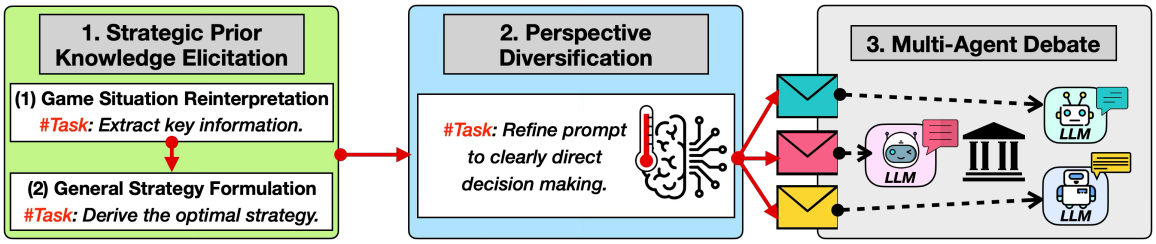


Figure 1: **DReaMAD** framework. **DReaMAD** improves LLM reasoning by combining **Strategic Prior Knowledge Elicitation** and **Perspective Diversification**. First, the model reinterprets the problem and formulates high-level strategies to reduce belief entrenchment. In the second stage, multiple agents adopt distinct viewpoints, engage in structured debate, and refine their conclusions to enhance decision-making.

prompts to foster diverse perspectives, thereby reshaping debate dynamics. Our key contributions are as follows:

- We identify and decompose **belief entrenchment** in Multi-Agent Debate, showing that standard MAD acts as a dynamic amplifier of static cognitive biases.
- We propose **DReaMAD**, a novel framework that sequentially targets the root causes of belief entrenchment through strategic knowledge elicitation and perspective diversification. It achieves a +9.5% accuracy gain over ReAct prompting on the *MetaNIM Arena* dataset and a +19.0% higher win rate than MAD in the simulator.
- We introduce **MetaNIM Arena**, a benchmark designed to evaluate LLMs in adversarial strategic decision-making, enabling precise assessment of reasoning robustness and strategic adaptability.

2 MetaNIM Arena

We introduce *MetaNIM Arena*, a benchmark designed to rigorously evaluate adversarial strategic reasoning in LLMs. Unlike static QA tasks, this benchmark requires agents to make sequential decisions in dynamic environments where optimal play is mathematically definable but non-trivial to execute.

2.1 Mathematical Foundation: Impartial Games

The arena consists of five *impartial games*: NIM, Fibonacci Nim, Kayles, Chomp, and Corner Queen. These games share key properties: (1) **Full Observability:** The entire game state is visible to both players; (2) **Impartiality:** Available moves depend solely on the state, not on which player is moving; (3) **Termination:** The game is guaranteed to end in a finite number of moves.

Crucially, for any non-terminal state in these games, one player holds a provably winning strategy assuming optimal play. This is grounded in the **Sprague-Grundy Theorem**, which maps every impartial game state to a non-negative integer called a *Grundy number* (or *nimber*). A state with a Grundy number of 0 is a losing position (P-position), while any non-zero state is a winning position (N-position). This provides a clear, objective criterion for optimality: an agent reasoning correctly should always move the game to a state with a Grundy number of 0.

2.2 Game Variants and Complexity

MetaNIM Arena includes diverse game settings to test different aspects of strategic reasoning:

- **NIM:** The foundational game where players remove objects from heaps. The winning strategy relies on the XOR sum of heap sizes (Nim-sum).
- **Fibonacci Nim:** A dynamic variant where the maximum removal is constrained by the opponent's previous move (up to $2\times$). This introduces complex state dependencies requiring forward lookahead.
- **Kayles:** Players knock down pins, potentially splitting a row into two independent subgames. This tests the agent's ability to decompose problems and analyze disjoint game components.
- **Chomp:** A geometric game on a grid where players consume blocks. While a winning strategy is guaranteed to exist, no simple closed-form solution is known for general grid sizes, making it a test of heuristic search and pattern recognition.
- **Corner Queen:** A spatial game equivalent to *Wythoff's game*, where winning positions follow the Beatty sequence (related to the Golden Ratio), requiring precise numerical reasoning.

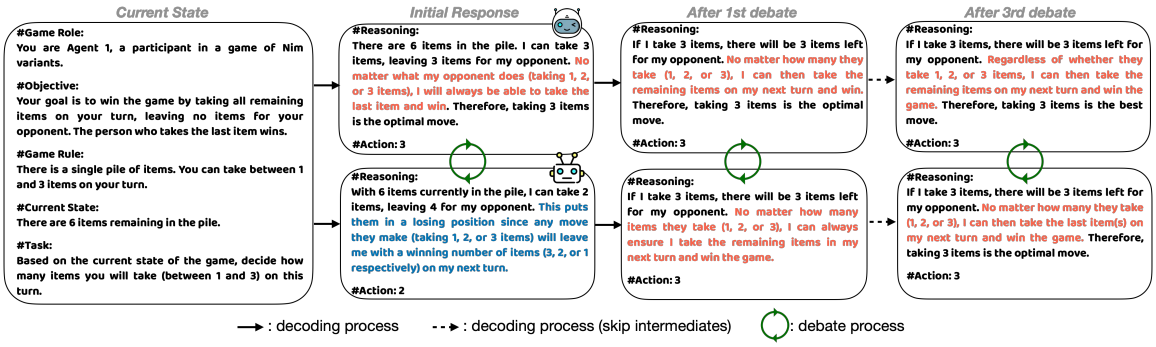


Figure 2: An example demonstrating how the debate process converges to a biased outcome. We observed that belief entrenchment occurs in the first debate. Blue text indicates the correct reasoning and orange text indicates the strong consistent (biased) reasoning. The second debate is omitted, as its procedure replicates the first and third; all debates use `GPT-4o-mini` as the debating agent.

We provide a more detailed theoretical explanation for impartial games and each game variant in Appendix A.1.

2.3 Evaluation Modes

We utilize *MetaNIM Arena* in two complementary modes:

Interactive Simulator. This mode evaluates an agent's adaptability in a turn-based environment. The agent plays against a strong adaptive opponent powered by `GPT-4o` using the ReAct framework (Yao et al., 2023). To ensure rigorous evaluation, the agent is always initialized in a winning position (N-position). Performance is measured by win rate, reflecting the agent's ability to maintain the winning advantage against an optimal or near-optimal adversary. We employ various configurations (e.g., different board sizes, starting items) as detailed in Appendix A.2.

Static Dataset. To analyze reasoning quality at a granular level, we constructed a static dataset of critical game states. Each entry consists of a state description and its unique optimal move (calculated via Grundy numbers). This allows us to measure **Decision Accuracy**, defined as the percentage of turns where the agent selects the theoretically optimal move, without the noise of full game simulations.

3 Analyzing Belief Entrenchment in Debate

Before introducing our method, we systematically analyze why standard Multi-Agent Debate (MAD) often fails to correct errors and instead reinforces inherent belief. We utilize the *MetaNIM Arena* environment to quantitatively measure this phenomenon.

3.1 Experimental Setup and Definitions

To rigorously analyze the debate dynamics, we focus on the *Fibonacci Nim* game, where the optimal move is mathematically deterministic. This allows us to clearly distinguish between the model's inherent preference and the ground-truth optimal action.

Estimating Argument Probability. Departing from token-level log-probabilities used in prior work, we estimate the probability p_i of an argument (action) i via empirical generation frequency. Specifically, we sample $N = 20$ responses at a fixed temperature ($T = 0.7$) to determine the action distribution. Note that p_i represents the model's belief in a specific context; it is not static but shifts based on the prompting strategy.

Defining Belief and Consistency. We conceptualize the model's *static belief* as its initial action probability distribution, denoted as $p_i^{(0)}$, prior to any multi-agent interaction.

- **Optimal Action:** The theoretically unique move that guarantees a winning strategy from the current state, serving as the ground truth for correctness.
- **Suboptimal Tendency:** A systematic preference for a losing action driven by heuristic shortcuts.
- **Strong Consistency:** A state where the model's belief is highly concentrated, specifically when the dominant action's probability exceeds the majority threshold ($p_i > 0.5$). This indicates a solidified stance likely to dominate the consensus process.

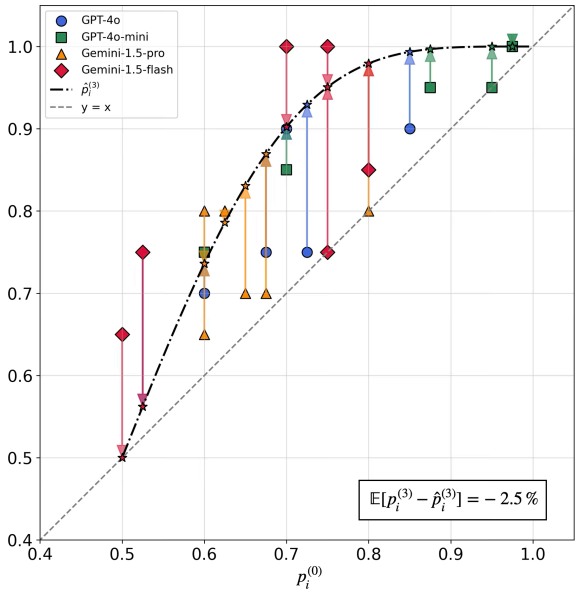


Figure 3: Belief entrenchment across models: showing that even after the debate concludes, strongly consistent actions continue to exhibit strong consistency, reinforcing biased action distributions in the Fibonacci game.

3.2 Empirical Evidence and Modeling

We conducted experiments using 3 rounds of standard MAD (Du et al., 2023) with various agents against `GPT-4o`. Figure 3 illustrates the change in probability of the dominant action after the debate.

Observation: Belief Entrenchment. As shown in Figure 3, actions that initially exhibited Strong Consistency ($p^{(0)} > 0.5$) consistently increased their probability after debate ($p^{(3)} > p^{(0)}$), regardless of whether the action was optimal or suboptimal. Furthermore, all data points for Strong Consistency lie above the $y = x$ line. This empirically demonstrates **belief entrenchment**: the debate process acts as an amplifier for the majority opinion, rather than a correction mechanism based on logical validity, as illustrated in Figure 2. This leads agents to converge on a suboptimal consensus, as shown in Figure 4, where a correct counterargument is overridden by the reinforced bias, leading the agents to converge on a suboptimal consensus.

Modeling the Dynamics. To explain the mechanism behind this phenomenon, we propose a probabilistic interaction model based on **pairwise agent debate**. We posit that the persuasiveness of an argument is not determined by its logical validity, but is proportional to the model's static initial belief, $p_i^{(0)}$. This implies that agents are more likely to accept arguments that align with their inherent biases.

Mathematically, the probability of action i at round $t+1$, denoted as $p_i^{(t+1)}$, evolves as follows:

$$p_i^{(t+1)} = \underbrace{p_i^{(t)} \cdot p_i^{(t)}}_{\text{Consensus}} + \underbrace{2p_i^{(t)}(1-p_i^{(t)}) \cdot p_i^{(0)}}_{\text{Belief-Driven Resolution}}, \quad (1)$$

where the first term represents the probability of both agents agreeing on action i, and the second term models conflict resolution biased by the initial popularity $p_i^{(0)}$.

Our simple model generates the theoretical curve in Figure 3, which closely fits the empirical data points with an average residual of just -2.5%. This tight alignment validates our hypothesis: the debate dynamics act as an amplifier of static bias ($p^{(0)}$) rather than a filter for reasoning quality.

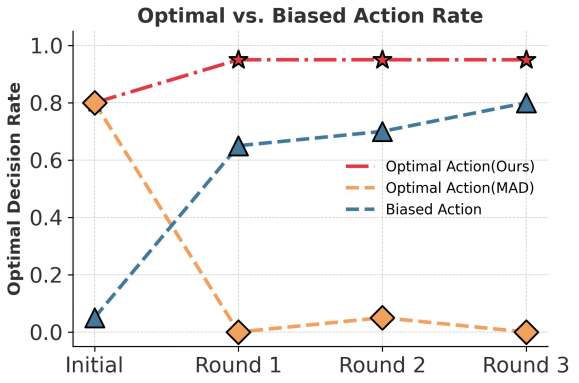


Figure 4: Belief Entrenchment in Action (NIM). Despite one agent proposing the optimal move, the debate converges to the suboptimal majority view, illustrating how valid reasoning is suppressed by the model's static belief.

Decomposing the Failure. Crucially, the strong fit of our model allows us to **decompose** belief entrenchment into two distinct root causes, motivating our proposed method:

1. **Static Initial Belief ($p^{(0)}$):** The model's inherent bias prior to any interaction. If the initial probability of the correct action is low ($p^{(0)}_{\text{correct}} < 0.5$), the debate cannot recover it.

2. **Homogenized Debate Dynamics:** The functional dependency where an argument's persuasiveness is dictated by its initial likelihood ($p^{(0)}$) rather than its logical validity. As evidenced by our model fit, this creates an echo chamber where homogeneous agents reinforce the dominant view regardless of its correctness.

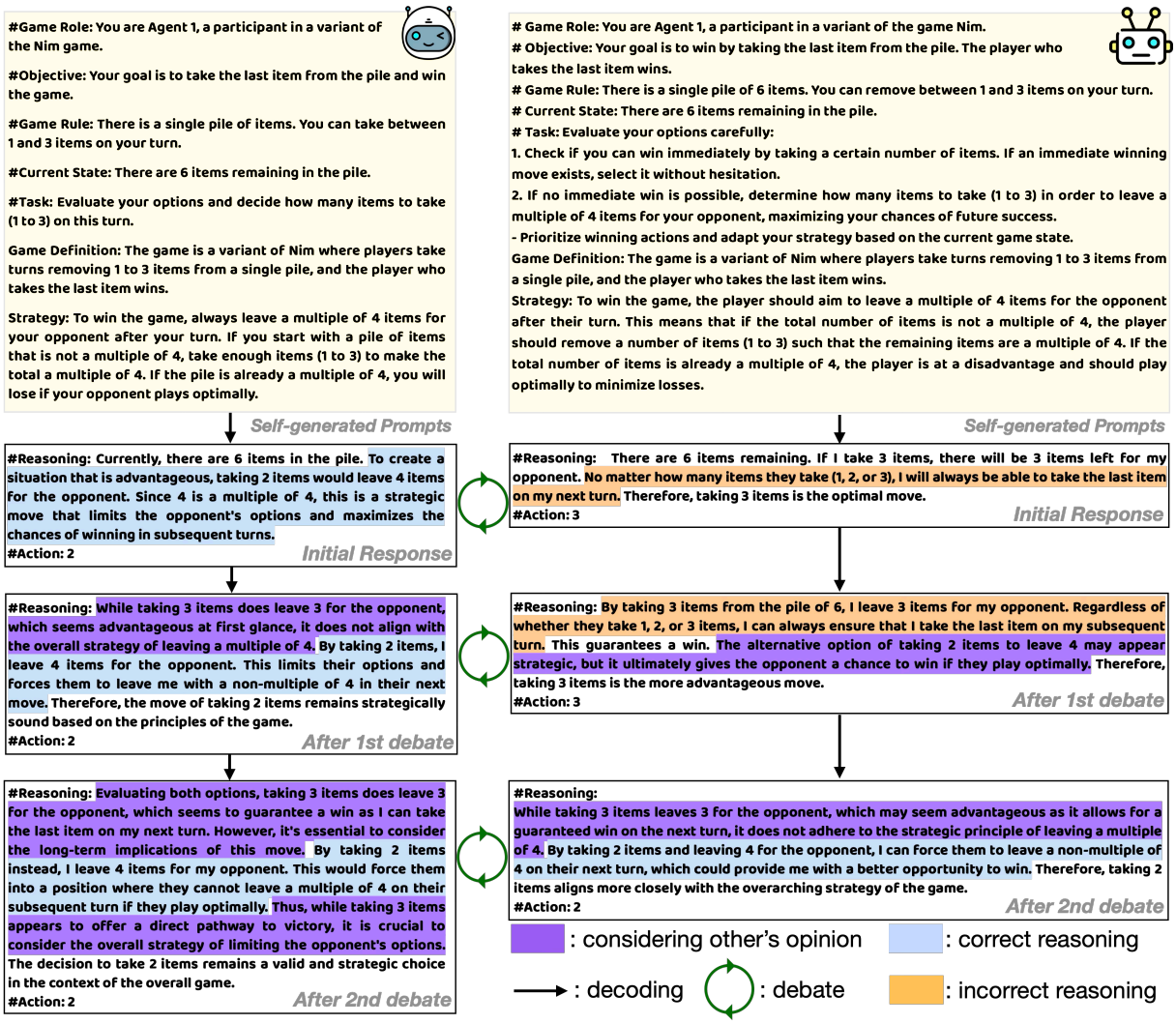


Figure 5: In this example, we illustrate how the debate process converges to an optimal outcome using our algorithm, **DReaMAD**. We begin with the same *current state* shown in Figure 2, employing self-generated prompts for each LLM agent.

4 DReaMAD: Counteracting Belief Entrenchment

To address the limitations of standard MAD identified in Section 3, we introduce **DReaMAD** (Diverse Reasoning via Multi-Agent Debate with Refined Prompt). Our framework counteracts belief entrenchment by targeting its two root causes: the initial static bias ($p^{(0)}$) and the homogenized debate dynamics.

4.1 Stage 1: Strategic Prior Knowledge Elicitation (SPKE)

The first stage aims to correct the model's initial reasoning bias before any debate begins. Standard prompting often leads models to rely on superficial heuristics, resulting in a skewed initial probability distribution $p^{(0)}$.

Method. We implement a two-step elicitation process. First, in *Game Situation Reinterpretation*, the model is prompted to analyze the problem's structure and constraints explicitly. Second, in *General Strategy Formulation*, it formulates high-level winning strategies. This step stops the model from locking into a wrong path early. We set the temperature to `0.1` to ensure consistent strategic output.

Theoretical Justification. By forcing the model to ground its reasoning in domain-specific strategic knowledge, SPKE shifts the initial probability mass $p^{(0)}$ towards the optimal action. According to our model in Eq. 1, increasing the initial probability of the correct argument is crucial because the subsequent debate dynamics tend to amplify the dominant initial belief.

LLM Models	Prompting Methods	NIM	Fibonacci	Chomp	Kayles	Average
GPT-4o	ReAct	0.95 ± 0.04	0.33 ± 0.04	0.18 ± 0.07	0.19 ± 0.08	0.41
	+ CoT-Prompting	0.96 ± 0.04	0.43 ± 0.11	**0.28** ± 0.09	0.20 ± 0.10	**0.47**
	DReaMAD$^{(-)}$	**0.98** ± 0.04	**0.44** ± 0.09	0.23 ± 0.10	**0.23** ± 0.12	**0.47**
GPT-4o-mini	ReAct	0.75 ± 0.05	0.33 ± 0.04	0.40 ± 0.07	0.12 ± 0.06	0.40
	+ CoT-Prompting	0.84 ± 0.08	0.36 ± 0.06	0.61 ± 0.05	0.02 ± 0.03	0.46
	DReaMAD$^{(-)}$	**1.00** ± 0.00	**0.49** ± 0.17	**0.62** ± 0.10	**0.18** ± 0.11	**0.57**
Gemini-1.5-pro	ReAct	0.82 ± 0.06	0.42 ± 0.04	0.19 ± 0.08	0.57 ± 0.04	0.50
	+ CoT-Prompting	0.88 ± 0.05	0.47 ± 0.11	0.22 ± 0.11	0.59 ± 0.04	0.54
	DReaMAD$^{(-)}$	**0.97** ± 0.04	**0.53** ± 0.07	**0.24** ± 0.05	**0.72** ± 0.12	**0.62**
Gemini-1.5-flash	ReAct	0.94 ± 0.02	0.35 ± 0.04	0.05 ± 0.03	0.01 ± 0.02	0.34
	+ CoT-Prompting	0.93 ± 0.02	0.33 ± 0.07	**0.09** ± 0.04	0.0 ± 0.00	0.34
	DReaMAD$^{(-)}$	**0.97** ± 0.04	**0.45** ± 0.06	0.05 ± 0.00	**0.42** ± 0.06	**0.46**
Qwen3-4B	ReAct	**0.99** ± 0.02	**0.42** ± 0.09	0.08 ± 0.05	0.47 ± 0.03	**0.49**
	+ CoT-Prompting	0.98 ± 0.02	0.38 ± 0.04	**0.08** ± 0.03	**0.48** ± 0.02	0.48
	DReaMAD$^{(-)}$	**0.99** ± 0.02	**0.42** ± 0.09	0.04 ± 0.02	0.47 ± 0.06	0.48
Average	ReAct	0.89	0.37	0.18	0.27	-
	+ CoT-Prompting	0.92	0.39	**0.26**	0.26	-
	DReaMAD$^{(-)}$	**0.98**	**0.47**	0.24	**0.40**	-

Table 1: Effect of Strategic Prior Knowledge Elicitation module. **DReaMAD**$^{(-)}$ indicates our method except multi-agent debate process. We can fully evaluate reasoning ability between different prompting methods. The metric accuracy of selecting optimal action is used. The best results are highlighted in **bold**.

4.2 Stage 2: Perspective Diversification

Even with a corrected initial belief, standard MAD suffers from an "echo chamber" effect where homogeneous agents reinforce each other's remaining biases. This corresponds to the persuasiveness being dependent on the argument's initial popularity rather than its quality.

Method. We break this dependency by assigning each agent a unique, self-generated prompt. Inspired by the *Learning from Multiple Approaches* theory (Council et al., 2005; Cleaves, 2008), this encourages agents to adopt distinct reasoning paths. For example, as shown in Figure 5, even starting from the same state, agents guided by diverse prompts generate distinct internal reasoning, preventing premature convergence to a wrong consensus. We set the temperature to `0.7` to promote diversity.

Theoretical Justification. This module directly targets the debate dynamics. By forcing agents to adopt different perspectives, we ensure arguments win based on logic, not popularity $p^{(0)}$. This ensures that arguments are evaluated based on their intrinsic logical merit rather than their alignment with the majority bias, fostering a robust, truth-seeking debate.

After these stages, the agents conduct a structured multi-agent debate. The complete workflow is illustrated in Figure 1 and the detailed prompt formulation used in these two modules is documented in the Appendix A.2.2.

5 Experiments

We utilized the benchmark *MetaNIM Arena* as both our dataset and simulator, as it provides a controlled environment for evaluating reasoning under grounded strategic tasks. Our investigation focuses on three key questions: (1) Does our approach improve reasoning quality compared to existing prompting techniques? (2) Does our approach prove its strategic reasoning quality under adversarial decision-making environments? (3) Does generating diverse prompts contribute to better decision-making within the debate framework? Further, we conduct two more ablation studies: (4) Can **DReaMAD** achieve long CoT reasoning without further training? (5) Can **DReaMAD** be generalized to general NLP domain?

To address these questions, we compare **DReaMAD** with standard prompting methods including ReAct (Yao et al., 2023), Zero-shot Chain-of-Thought (CoT), Self-Consistency (Wang et al., 2022), Self-Refinement (Madaan et al., 2024), and Multi-Agent Debate (MAD (Du et al., 2023) and Multi-Persona Debate (MPD) (Liang et al., 2023)). The details of standard and CoT prompts are provided in Appendix A.2.2.

Our method builds on the MAD framework by Du et al. (2023), augmenting it with structured self-prompt refinement and perspective diversification.

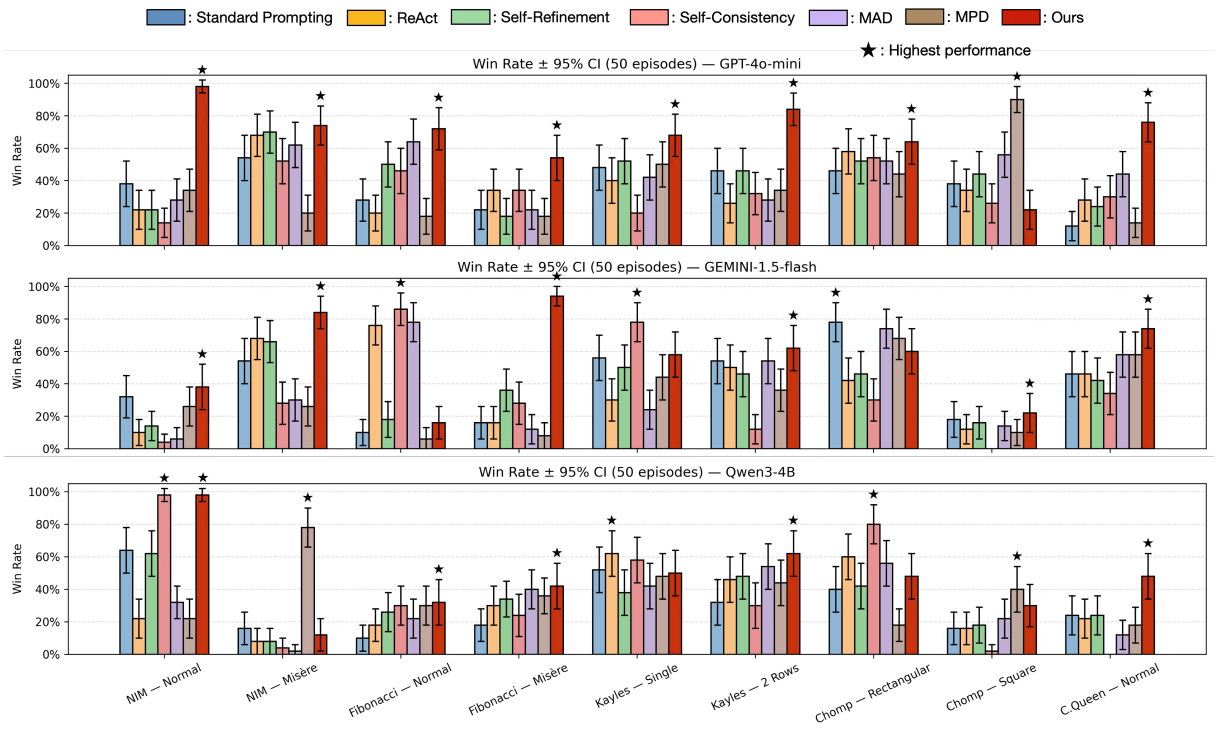


Figure 6: Winning rate comparison across different models and different self-correction methods. This is a result based on *MetaNIM Arena* simulator. The best results are highlighted in **bold**. We report win rates with 95% confidence intervals over 50 episodes.

For self-refinement, we follow the methodology of Madaan et al. (2024), applying three iterative refinement steps. Similarly, for MAD, we conducted up to three rounds of debate, following Du et al. (2023), with the process terminating early if a consensus is reached before the final round. We also investigate whether the observed improvements generalize across different LLM architectures, including `GPT` (Achiam et al., 2023), `Gemini` (Team et al., 2023), and `Qwen` (Yang et al., 2025) models.

5.1 Does DReaMAD Improve Reasoning Quality?

This experiment isolates the effect of strategic prior knowledge elicitation, allowing us to assess whether our method enhances decision-making without relying on debate dynamics.

Setup. To evaluate the effectiveness of our approach in improving reasoning capabilities, we compare **DReaMAD** without the debate process against ReAct and Zero-shot CoT prompting across multiple models in the *MetaNIM Arena* dataset (§A.2). For showing versatility of **DReaMAD**, we conduct experiments on five variants of LLMs as shown in Table 1.

Results. Table 1 demonstrates that **DReaMAD**$^{(-)}$ generally improves performance over ReAct and CoT prompting across all models and tasks. These results highlight the impact of our method in reinforcing structured strategic reasoning, even without the iterative correction process of debate. Notably, our approach leads to substantial improvements in NIM, Fibonacci, and Kayles, which are environments where long-term strategic planning plays a crucial role. Since defining a general winning strategy in Chomp is non-trivial, applying prior knowledge is challenging and results in less effectiveness compared to other games. Furthermore, the gains tend to be larger for some weaker baselines (e.g., `GPT-4o-mini` by +17% and `Gemini-1.5-flash` by +12% on average), whereas very small models such as `Qwen3-4B` show little to no benefit, likely due to limited capacity of prompts refinement and eliciting priors.

5.2 DReaMAD in Adversarial Strategic Decision-Making

Setup. We applied **DReaMAD** to `GPT-4o-mini`, `Gemini-1.5-flash`, and `Qwen3-4B` then compared it with self-correction methods, including standard-prompt, ReAct, self-refinement, self-consistency, MAD and MPD. To demonstrate its effectiveness, we used `GPT-4o` as the opponent model due to its

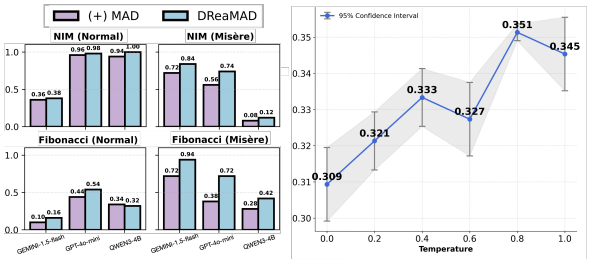


Figure 7: Effect of perspective diversification. Left: Average win rate of **(+)MAD** and **DReaMAD** on NIM and Fibonacci (Normal and Misère variants), aggregated over 50 simulations per setting. Right: Accuracy on the Fibonacci benchmark with `GPT-4o` across different sampling temperatures (15 runs each, 95% CI). Higher temperatures yield greater prompt diversity, leading to improved accuracy.

superior performance. In these experiments, we utilized *MetaNIM Arena* simulator to maximize the effect of generating diverse prompts. We aimed to validate our hypothesis in a simulator that requires strategic decision-making within complex dynamics. We ran 50 independent episodes and average the win-rate.

Results. As shown in Figure 6, **DReaMAD** consistently outperforms other self-correction methods across various strategic environments, demonstrating a significant improvement in winning rates. This result suggests that our approach enables LLM agents to effectively adapt to complex dynamics, particularly in adversarial decision-making scenarios where strategic reasoning is crucial. Notably, even for a small model such as `Qwen3-4B`, **DReaMAD** achieves the best overall performance among the compared methods; however, the magnitude of improvement tends to be larger for higher-capacity models, indicating that additional model capacity better translates elicited strategic priors into effective decision policies. However, we observe that **DReaMAD** struggles in the Chomp game, which aligns with the fact that, although a winning strategy is guaranteed to exist for the first player by the strategy-stealing argument, no general closed-form solution is known for arbitrary board sizes. This proof establishes the existence of a winning strategy without explicitly constructing it, making the task highly non-trivial in practice. The absence of a fully characterized optimal strategy means that effective play often requires exploratory search rather than direct reasoning from prior knowledge. This highlights a limitation of our method in environments where strategic heuristics are only partially understood or insufficiently structured.

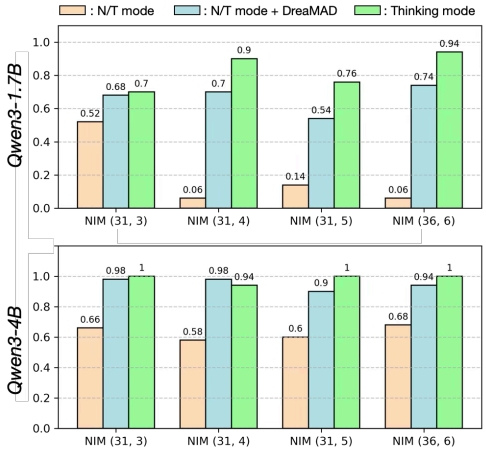


Figure 8: Performance gain of **DReaMAD** when applied to the Non-thinking (N/T) mode. The figure shows the average win rate over 50 runs for the `Qwen3-1.7B` (top) and `Qwen3-4B` (bottom) models. The notation NIM (n, k) indicates a game starting with n stones and a maximum take of k stones per turn.

5.3 Does Generating Diverse Prompts Improve Performance?

We assess whether prompt diversity improves decision quality within the MAD framework. To this end, we compare two settings: (1) identical prompts generated via the Strategic Prior Knowledge Elicitation module for both agents as **(+)MAD**, and (2) distinct, self-generated prompts per agent, as in **DReaMAD** (Figure 7, left). Experiments were conducted on four variants of the *MetaNIM Arena* simulator, NIM (Normal and Misère) and Fibonacci (Normal and Misère), over 50 episodes each. Our results show that incorporating diverse prompts within the MAD framework significantly enhances performance, validating the effectiveness of our Perspective Diversification module.

We also examine the effect of sampling temperature on prompt diversity in the Fibonacci task. Within both the Strategic Prior Knowledge Elicitation and Perspective Diversification modules, we vary the temperature from 0.0 to 1.0. As shown in Figure 7 right, higher temperature (further diversity) correlates with increased optimal action accuracy, indicating that greater diversity in generated prompts contributes to improved reasoning performance. However, we also note that excessively high temperatures could risk degrading reasoning coherence by introducing noise. Determining the optimal temperature that balances diversity and logical consistency remains an important consideration for practical applications.

Dataset	GPT-o3-mini					GPT-4o				
	ReAct	Self-Refine.	Self-Consist.	MAD	**DReaMAD**	ReAct	Self-Refine.	Self-Consist.	MAD	**DReaMAD**
AIME 2024	72.0 ± 4.47	74.0 ± 1.49	**82.7** ± 2.79	**82.7** ± 2.79	**82.7** ± 2.79	8.67 ± 4.47	8.67 ± 2.98	10.7 ± 2.71	11.3 ± 5.06	**12.0** ± 5.58
AMC 2023	99.0 ± 1.37	98.0 ± 1.12	99.5 ± 1.12	99.5 ± 1.12	**100** ± 0.0	51.5 ± 3.29	54.5 ± 4.81	58.0 ± 5.42	55.0 ± 5.59	**59.5** ± 4.47

Table 2: Accuracy (%) on math-reasoning benchmarks. We report the mean and standard deviation over 5 independent runs. Bold indicates the best algorithm for a given pair. While DReaMAD consistently achieves the highest mean accuracy, the overlap in standard deviations with Self-Consistency suggests comparable performance in some settings, highlighting the robustness of ensemble-based methods.

5.4 Achieving Long CoT Reasoning Without Specialized Training

Researchers often assume that long CoT reasoning requires costly post-training, such as the multi-stage reinforcement learning pipeline used to create `Qwen3`'s thinking mode. In this section, we investigate a critical question: can the latent reasoning capabilities of a base model (non-thinking mode) be unlocked to match this long CoT reasoning performance without undergoing such training?

To explore this, we analyze the performance of the `Qwen3-1.7B` and `4B` model (Yang et al., 2025) on 4 variants of NIM-Normal game which have different initial number of stones and maximum number of stones that can be taken under three conditions, as shown in Figure 8: (1) the standard, Non-thinking (N/T) mode; (2) the heavily post-trained Thinking mode; and (3) **DReaMAD** on top of the N/T mode.

The results reveal the substantial value gained by our approach, demonstrating its effectiveness across different model scales. The standard Non-thinking mode models establish a performance baseline, which our framework significantly improves upon. For the `Qwen3-1.7B` model, **DReaMAD** achieves a remarkable 226% average performance gain, elevating a weak baseline to 81.3% of the specialized Thinking mode's capability.

This effect is even more pronounced on the more capable `Qwen3-4B` model. Here, applying **DReaMAD** yields a 45.1% average performance gain, bringing the base model to 97.9% of the Thinking mode's performance and thus achieving near-parity: while the Thinking mode still attains the highest raw accuracy, **DReaMAD** recovers most of its benefit without any additional post-training. The primary gain is one of efficiency and accessibility; **DReaMAD** provides a training-free pathway to elicit long CoT reasoning from general-purpose models at test-time, bypassing the need for separate, resource-intensive training pipelines.

Model	ReAct	Self-Refine.	Self-Consist.	MAD	**DReaMAD**
`GPT-4o`	83.93 ± 1.24	83.11 ± 1.37	**84.75** ± 0.93	84.26 ± 1.47	83.93 ± 1.10
`GPT-4o-mini`	77.38 ± 1.37	75.25 ± 3.95	76.89 ± 0.69	77.38 ± 0.73	**78.85** ± 1.07

Table 3: Accuracy (%) on CommonsenseQA dataset. We report the mean and standard deviation over 5 runs. Bold indicates the best performance. DReaMAD shows a statistically significant improvement over ReAct for `GPT-4o-mini`, while performing comparably to Self-Consistency for `GPT-4o`.

5.5 Generalization to Math and Common Sense Reasoning

While our primary benchmark focuses on structured games, we further evaluate **DReaMAD** on NLP tasks to test its broader applicability. Specifically, we consider algebra and number theory problems from AIME 2024 and AMC 2023, as well as CommonsenseQA (Talmor et al., 2018). Experiments are conducted using standard accuracy metrics across model-task pairs, averaged over 5 independent runs.

Results in Table 2 and Table 3 show that **DReaMAD** remains competitive or superior beyond our MetaNIM Arena benchmark. In math reasoning (Table 2), **DReaMAD** clearly improves over ReAct and Self-Refine. For example, on AIME 2024 with `GPT-o3-mini` it reaches 82.7%, about 9–11 points higher than ReAct (72.0%) and Self-Refine (74.0%), and on AMC 2023 it attains 100% accuracy, slightly surpassing Self-Consistency. For `GPT-4o`, **DReaMAD** also gives the highest means on both AIME 2024 and AMC 2023, though the confidence intervals overlap with Self-Consistency and MAD, indicating comparable robustness.

Similarly, in CommonsenseQA (Table 3), **DReaMAD** achieves the highest mean accuracy for `GPT-4o-mini` (78.85%), demonstrating statistically significant gains over baselines. For the stronger `GPT-4o` model, it performs on par with Self-Consistency. These findings align with our observations in Section 5.4 (`Qwen` analysis): while specialized training (e.g., Thinking mode) sets a

high baseline, **DReaMAD** offers a training-free pathway to unlock latent reasoning capabilities, achieving long CoT performance or further refining the outputs of reasoning-specialized models.

6 Related Works

6.1 Cognitive Bias and Belief Entrenchment in LLMs

While the term *bias* in Large Language Models (LLMs) frequently refers to societal or demographic disparities (Gallegos et al., 2024), our work investigates *cognitive biases*, which are systematic deviations from rational decision-making that hinder robust reasoning. Specifically, we focus on **belief entrenchment**, a phenomenon where models exhibit excessive confidence in their initial, often erroneous, outputs and resist correction.

Recent studies suggest that LLMs often prioritize consistency over accuracy. This tendency arises because models are trained to maximize the likelihood of the next token based on context, leading them to align responses with user prompts or their own previous outputs rather than objective truth (Perez et al., 2023). Furthermore, Turpin et al. (2023) demonstrated that Chain-of-Thought (CoT) reasoning is frequently biased by the model's initial prediction. Instead of reasoning to find the answer, the model generates explanations to justify its initial choice. This aligns with our finding that models tend to reinforce their **static belief**, defined as the initial probability distribution over actions, regardless of its correctness.

This belief entrenchment is further intensified by the model's reliance on consistency as a proxy for correctness. While Wang et al. (2022) showed that aggregating consistent reasoning paths generally improves performance, this approach fails when the model is confident in a wrong answer, as shown. In such cases, the model consistently generates plausible but incorrect responses. Although models possess some capability to evaluate the validity of their own claims (Kadavath et al., 2022), this internal calibration often fails to override the reinforced consistency in iterative settings. Consequently, methods like Self-Refinement often fail to fix errors (Huang et al., 2024; Valmeekam et al., 2022), instead creating "reasoning loops" that amplify the model's inherent biases. Our work extends these observations to the multi-agent setting, demonstrating that without structural intervention to diversify perspectives, Multi-Agent Debate (MAD) acts as a dynamic amplifier of these static cognitive biases, effectively forming an echo chamber of homogeneous agents.

6.2 Prompt Engineering and Self-Correction in LLMs

Prompt engineering shapes model outputs without retraining, potentially improving generalization and reducing bias (Brown et al., 2020; Reynolds and McDonell, 2021; Zhao et al., 2024; Schulhoff et al., 2024; Shin et al., 2024). However, fully eliminating biases in complex reasoning remains challenging (Jiang et al., 2022; Lu et al., 2022).

Meanwhile, self-correction mechanisms in LLMs refine responses without external supervision (Ganguli et al., 2023; Liu et al., 2024; Kamoi et al., 2024). Self-consistency, for instance, ensembles multiple outputs but converges on frequent rather than correct answers (Wang et al., 2022; Chen et al., 2023), and self-refinement can reinforce rather than fix biases (Wan et al., 2023; Shinn et al., 2023; Madaan et al., 2024; Huang et al., 2024). Feedback-loop methods such as STaR (Zelikman et al., 2022), Reflexion (Shinn et al., 2023), and SCoRe (Kumar et al., 2024) also struggle to reliably correct biases or foster diverse reasoning (Guo et al., 2024).

6.3 Multi-Agent Debate in LLMs

Multi-Agent Debate (MAD) enables LLM agents to critique each other, enhancing reasoning on complex tasks (Liang et al., 2023; Du et al., 2023). ChatEval (Chan et al., 2023), a multi-agent evaluation system, simulates human judgment to assess model output quality. Optimizations include task-specific strategies for improving debate effectiveness (Smit et al., 2024) and ACC-Debate, an actor-critic framework that trains models to specialize in debates, achieving benchmark gains (Estornell et al., 2024). Highlighting similar risks, (Estornell and Liu, 2024) provided a theoretical framework demonstrating that multi-LLM debates are susceptible to "echo chamber" effects, where a flawed majority opinion arising from shared misconceptions can suppress correct minority viewpoints. While these enhancements improve performance, studies reveal a key limitation: static evaluations focus on assessing predefined problems, whereas real-world decision-making often involves dynamic, interactive environments where biases can evolve. Understanding how biases shift in these settings is crucial for developing robust strategies that extend beyond conventional static benchmarks.

7 Discussion

Limitation. While our approach demonstrates consistent improvements across a variety of strategic reasoning tasks, it is not without limitations. For instance, in the game of *Chomp*, a winning strategy is guaranteed for the first player by the strategy-stealing argument, but no general, closed-form solution is known. Our framework, which relies on eliciting and refining prior strategic knowledge, is consequently less effective in such scenarios. This was reflected in our experiments, where performance on the Chomp dataset and simulator was significantly lower than on other games (Table 1 and Figure 6). Chomp aside, we also observe that while our method improves performance on *CommonsenseQA*, the gains are relatively modest compared to those on strategic and mathematical tasks (Table 2). This highlights that our method excels when a strong strategic foundation can be leveraged.

Future Directions. A key future direction involves comparing our single-model approach with heterogeneous model ensembles. While employing different models could inherently increase perspective diversity, this often introduces significant practical overhead. Our work prioritizes a more tractable solution by eliciting diversity from a single model, a crucial step for real-world deployment. Systematically evaluating the trade-offs between these two strategies, and potentially integrating them, remains a promising area for future research.

8 Conclusion

Our study shows that Multi-Agent Debate (MAD) often reinforces biases instead of reducing them, leading to suboptimal reasoning. Through our experiments with the *MetaNIM Arena*, we have observed that models persist in biased reasoning even when presented with superior alternatives. While our current strategy focuses on strategic games, the principles of structured self-refinement and diversified reasoning could be valuable for a wider range of NLP tasks. These include complex activities such as multi-step reasoning in question answering, legal analysis, and scientific inference. Future work will explore how these techniques enhance decision-making beyond structured games.

Acknowledgments

This work was supported by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD230017TD).

We sincerely thank the action editor, Antoine Bosselut, for their guidance and careful handling of the review process. We also thank the anonymous reviewers for their thoughtful and constructive feedback, which substantially improved the clarity and quality of this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Wendy Pelletier Cleaves. 2008. Promoting mathematics accessibility through multiple representations jigsaws. *Mathematics Teaching in the Middle School*, 13(8):446–452.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

National Research Council, Suzanne Donovan, John Bransford, et al. 2005. *How students learn*. National Academies Press Washington, DC.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Steven J Edwards. 1994. Portable game notation specification and implementation guide. *Retrieved April*, 4:2011.

Andrew Estornell and Yang Liu. 2024. Multi-llm debate: Framework, principals, and interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964.

Andrew Estornell, Jean-Francois Ton, Yuanshun Yao, and Yang Liu. 2024. Acc-debate: An actor-critic approach to multi-agent debate.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. 2023. The capacity for moral self-correction in large language models.

P. M. Grundy. 1939. Mathematics and games. *Eureka*, 2:6–8.

Yufei Guo, Muzhe Guo, Juntao Su, Zhou Yang, Mengqiu Zhu, Hongfei Li, Mengyang Qiu, and Shuo Shuo Liu. 2024. Bias in large language models: Origin, evaluation, and mitigation.

Jie He, Tao Wang, Deyi Xiong, and Qun Liu. 2020. The box is in the pen: Evaluating commonsense reasoning in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3662–3672.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Liwei Jiang, Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny Liang, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jon Borchardt, Saadia Gabriel, Yulia Tsvetkov, Oren Etzioni, Maarten Sap, Regina Rini, and Yejin Choi. 2022. Can machines learn morality? the delphi experiment.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker,

Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Guangliang Liu, Haitao Mao, Bochuan Cao, Zhiyu Xue, Xitong Zhang, Rongrong Wang, Jiliang Tang, and Kristen Johnson. 2024. On the intrinsic self-correction capability of llms: Uncertainty and latent concept.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, Andy Jones, Anna Chen, Benjamin Mann, Brian Israel, Bryan Seethor, Cameron McKinnon, Christopher Olah, Da Yan, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Guro Khundadze, Jackson Kernion, James Landis, Jamie Kerr, Jared Mueller, Jeeyoon Hyun, Joshua Landau, Kamal Ndousse, Landon Goldberg, Liane Lovitt, Martin Lucas, Michael Sellitto, Miranda Zhang, Neerav Kingsland, Nelson Elhage, Nicholas Joseph, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robin Larson, Sam McCandlish, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Jack Clark, Samuel R. Bowman, Amanda Askell, Roger Grosse, Danny Hernandez, Deep Ganguli, Evan Hubinger, Nicholas Schiefer, and Jared Kaplan. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13387–13434, Toronto, Canada. Association for Computational Linguistics.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Pham, Gerson Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Hoyle, and Philip Resnik. 2024. The prompt report: A systematic survey of prompting techniques.

Philip Wootaek Shin, Jihyun Janice Ahn, Wenpeng Yin, Jack Sampson, and Vijaykrishnan Narayanan. 2024. Can prompt modifiers control bias? a comparative analysis of text-to-image generative models.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Andries Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D. Barrett, and Arnu Pretorius. 2024. Should we be going mad? a look at multi-agent debate strategies for llms. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

R. P. Sprague. 1935. Über mathematische kampfspiele. *Tôhoku Mathematical Journal*, 41:438–444.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965. Curran Associates, Inc.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2022. Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change). In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Yuxuan Wan et al. 2023. Self-polish: Enhance reasoning in large language models via problem refinement. *arXiv preprint arXiv:2305.14497*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter. 2024. Enhancing zero-shot chain-of-thought reasoning in large language models through logic. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6144–6166, Torino, Italia. ELRA and ICCL.

A MetaNIM Arena

Algorithm 1 MetaNIM Arena: Turn-Based Opponent Task with Two Agents

Require: Initial State S_0, Goal Condition G, Agents A_1 and A_2
Ensure: Final State S_f and Outcome

$t \leftarrow 0$ *Initialize turn counter*
$S \leftarrow S_0$ *Set initial state*
while $S \notin G$ **and** game is not terminated **do**
  if $t \bmod 2 = 0$ **then**
    $a_t \leftarrow A_1(S)$ *Agent 1's turn, selects action* a_t
  else
    $a_t \leftarrow A_2(S)$ *Agent 2's turn, selects action* a_t
  end if
  $S \leftarrow \text{UpdateState}(S, a_t)$ *Apply the action and update state*
  $t \leftarrow t + 1$ *Increment turn counter*
  if $S \in G$ **then**
    Success: Goal Achieved
  end if
end while

A.1 Background on Impartial Games

Games in the MetaNIM Arena are impartial games. We begin by outlining the relevant theoretical foundation, starting with the Sprague–Grundy theorem. In Table 4, we present the overall rules of the Nim game and its optimal strategies.

A.1.1 Theory and Strategy

All *MetaNIM Arena* games are impartial games, forming a Directed Acyclic Graph (DAG) where vertices represent game states and edges denote valid moves. The Grundy Number framework, along with the Sprague-Grundy Theorem, guarantees the existence of a winning strategy and provides a concrete method to determine it. In the *MetaNIM Arena*, each state has a mathematically provable optimal move, allowing an LLM's decisions to be evaluated against the *theoretical optimal strategy*—a key advantage for unbiased assessment.

Definition A.1 (Grundy Number)**.** For a finite impartial game under normal play (where the last player to make a valid move wins), the *Grundy number* (or *Nimber*) $G(S)$ is recursively defined as follows. If S is a *terminal state* with no valid moves, set $G(S) = 0$. Otherwise,

$$G(S) \;=\; \operatorname{mex}\{\, G(S') \mid S' \text{is reachable from } S \,\}.$$

Here, $\operatorname{mex}(X)$ is the smallest nonnegative integer not in X.

Note that the Grundy number is well-defined for every impartial game, since the game's state space forms a DAG. In many impartial games, direct enumeration of all possible move sequences is computationally infeasible. However, with Sprague-Grundy Theorem, we can easily calculate Grundy Numbers on complex games. See Appendix A.1.2 for further discussions.

Optimal Strategy. When $G(S) \neq 0$, there is *at least one* move to a successor S' with $G(S') = 0$, forcing the opponent into a losing position. Conversely, if $G(S) = 0$, *all* successor states have $G(S') \neq 0$. Because the game DAG is finite and acyclic, repeatedly applying "move to $G = 0$" (or avoiding it) ensures a *forced* result under optimal play. See Appendix A.1.3 for more details, including the *misère* variant where taking the last object *loses*.

A.1.2 Sprague-Grundy Theorem

The Sprague-Grundy theorem provides a fundamental method for analyzing impartial games by decomposing complex games into simpler, independent subgames. As discussed in Section A.1.1, every impartial game can be represented as a directed acyclic graph (DAG). However, directly computing Grundy numbers recursively from terminal states is often impractical.

We summarize key results from Sprague (1935) and Grundy (1939). According to the theorem, the optimal strategy for playing multiple impartial games simultaneously (in parallel), or a single complex game viewed as multiple independent subgames, is equivalent to playing a single game of Nim with multiple heaps. This equivalence arises from the concept of the disjunctive sum of DAGs.

Definition A.2 (Disjunctive Sum of DAGs)**.** Let $\mathcal{G}_1 = (X_1, F_1), \mathcal{G}_2 = (X_2, F_2), \ldots, \mathcal{G}_n = (X_n, F_n)$ be DAGs representing n impartial games. The disjunctive sum of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is a DAG $\mathcal{G} = (X, F)$ defined as follows:

1. The vertex set X is the Cartesian product $X_1 \times X_2 \times \cdots \times X_n$.
2. The edge set F consists of edges connecting $(x_1, \ldots, x_n)$ to $(y_1, \ldots, y_n)$ if and only if ex-

actly one pair (x_i, y_i) is in F_i, and $x_j = y_j$ for all $j \neq i$.

Note. In a disjunctive sum of DAGs, each player chooses exactly one subgame to play during their turn and moves within that subgame. The entire game ends when all subgames reach terminal positions.

Theorem A.3 (Sprague-Grundy (Sprague, 1935; Grundy, 1939))**.** *A position S is losing if and only if its Grundy number $G(S) = 0$; otherwise, if $G(S) \neq 0$, it is winning. Furthermore, if a position S decomposes into independent subpositions $S_1, \ldots, S_k$ via the disjunctive sum of DAGs, then*

$$G(S)_{(2)} = G(S_1)_{(2)} \oplus G(S_2)_{(2)} \oplus \cdots \oplus G(S_k)_{(2)},$$

where $\oplus$ denotes bitwise XOR.

This result implies that Grundy numbers for complex games, such as Kayles or Chomp, can be efficiently computed by decomposing them into simpler subgames and combining the Grundy numbers using bitwise XOR. For example, consider a variant of the game Kayles played on two separate rows of pins, each forming an independent subgame. Suppose we computed the Grundy numbers separately for these rows, obtaining Grundy numbers 7 for the first row and 4 for the second row. By the Sprague-Grundy theorem, the combined Grundy number of the position is given by:

$$7_{(2)} \oplus 4_{(2)} = 111_{(2)} \oplus 100_{(2)} = 011_{(2)} = 3.$$

Thus, even though the original game involves two distinct rows of pins, the strategic analysis reduces precisely to analyzing a Nim heap of size 3. Since a Nim heap of size 3 is nonzero, this indicates a winning position for the player about to move.

A.1.3 Basic Discussions on the Optimal Strategy

Why $G(S) = 0$ Implies Losing. Recall that $G(S)$ is defined as:

$$G(S) = \operatorname{mex}\Big\{ G(S') \;\Big|\; S' \text{is reachable from } S \Big\},$$

where $\operatorname{mex}(X)$ is the smallest nonnegative integer *not* in the set X. Thus,

$$G(S) = 0 \iff 0 \notin \{G(S') \mid S' \in \mathcal{R}(S)\}$$

where $\mathcal{R}(S)$ is the set of all states reachable from S. Concretely, if $G(S) = 0$, then *no valid move* leads to a successor S' with $G(S') = 0$. In other words, from S, the player to move *cannot* transition the game into a $G(\cdot) = 0$ state. Because a state $G(S') = 0$ corresponds to a losing position *for the player who faces it*, the mover in state S has *no way* to force the opponent into a losing position on the next turn. Hence, S is losing for the player to move.

Why $G(S) \neq 0$ Implies Winning (Opposite viewpoint). By the same logic, if $G(S) \neq 0$, then the definition of mex guarantees 0 *does* appear among the Grundy values $G(S')$ of the successors. Thus, there *exists* some child state S' for which $G(S') = 0$. Consequently, the current mover can place the opponent directly into a losing position (i.e. a position with Grundy number 0). Recursively iterating this argument along the Directed Acyclic Graph of states ensures that the current mover, if playing optimally, keeps forcing the opponent into $G(\cdot) = 0$ states until the game ends. Therefore, S must be a *winning* state.

Misère Variant. *Misère* play reverses the normal condition: taking the last object loses rather than wins. Although standard Sprague-Grundy analysis still applies to most states, a special exception arises when all heaps (or subpositions) are size 1, such as in *misère* Nim. In that endgame scenario, the usual strategy must switch to avoid forcing the final move, ensuring the player leaves the opponent to pick the last object.

A.2 Dataset and Simulator

We construct a simple dataset and a simulator based on the *MetaNIM Arena*. The dataset focuses on specific scenes in each game and does not require an opponent model, while the simulator enables interaction with an opponent that receives prompts and outputs actions, where we employ the GPT-4o model as the opponent.

A.2.1 Game Prompts

We provide how our MetaNIM Arena game's (NIM, Fibonacci, Chomp, Kayles, Corner-Queen) prompts are constructed as shown in Table 9. Any users can modify the games as they want to make.

A.2.2 Reasoning and `DReaMAD` Prompts

Standard, ReAct & CoT Prompts. In our evaluation of LLMs within the *MetaNIM Arena,* we compare several key prompting techniques: Standard, Zero-shot Chain-of-Thought (CoT), ReAct Prompting and **`DReaMAD`** as shown in Table 8. The dis-

Game	Rule	Mathematical Strategy
NIM	• k heaps; remove from exactly one heap • At least 1 up to a fixed maximum • Last move wins (normal play)	**Core:** Nim-sum $= n_1 \oplus \cdots \oplus n_k$ • **Winning:** Nim-sum $\neq 0$ (move to 0) • **Losing:** Nim-sum $= 0$ • Single-heap case $\Rightarrow$ modular arithmetic simplification
Fibonacci Nim	• Start with n items • First move: $1 \leq k < n$ • Next: $1 \leq x \leq 2\times$ (previous move) • Last move wins	**Core:** Governed by Fibonacci numbers and Zeckendorf's theorem • Zeckendorf: $m = F_{k_1} + F_{k_2} + \cdots + F_{k_r}, \lvert k_i - k_j \rvert \geq 2$ • **Winning:** Reduce to nearest smaller Fibonacci F_j • **Losing:** Pile size $= F_n$ • Optimal first move: remove the *smallest* Fibonacci in decomposition
Kayles	• Row of n pins • Knock down 1 pin or 2 adjacent pins • Last move wins	**Core:** Grundy numbers via Sprague–Grundy theorem • Recurrence: $G(n) = \text{mex}\{G(n-1), G(n-2), G(a) \oplus G(b)\}, a+b = n-k, k \in \{1,2\}$ • **Winning:** Grundy $\neq 0$ • **Losing:** Grundy $= 0$; periodic mod 12 when $n \geq 70$ • Strategic principles: split into XOR-sum $= 0$, mirror symmetric moves, avoid isolated single pins
Chomp	• $m \times n$ chocolate grid, (1,1) poisoned • Choose square; remove all above/right • Last move wins	**Core:** Strategy-stealing $\Rightarrow$ first player always wins • **Winning:** Guaranteed for first player (though constructive proof unknown) • **Losing:** Not characterized except for small grids • Analysis via posets under component-wise order • Strategic principles: control antidiagonal, enforce symmetry, reduce to subgames, avoid isolated columns
Corner Queen	• Queen at (x, y) on $m \times n$ grid • Move: left, down, or diagonal left-down by $k > 0$ • Win by reaching $(0, 0)$	**Core:** Equivalent to Wythoff's Game; Beatty sequence structure • Losing (P-positions): $(\lfloor k\phi \rfloor, \lfloor k\phi^2 \rfloor)$, $\phi = \frac{1+\sqrt{5}}{2}$ • Examples: $(1,2), (3,5), (4,7), (6,10), (8,13), (9,15), \ldots$ • **Winning:** Move to nearest P-position

Table 4: MetaNIM Arena: Detailed Rules and Mathematical Strategies of Game Variants

Methods	NIM	Fibonacci	Chomp	Kayles
Action space	1–3	dynamic (max 30)	x, y coordinate (scenario-dependent)	single pin or two adjacent pins
Variants	Normal	Normal	Square (2×2 – 19×19)	Normal
# samples	20	11	20	18

Table 5: Constructed dataset using *MetaNIM Arena*. We evaluate models on this dataset and report results in Table 1.

Features	NIM		Fibonacci	
	Normal	Misère	Normal	Misère
Starting Point	remaining items: 31	remaining items: 31	remaining items: 20	remaining items: 20
Winning Condition	taking last item	avoiding last item	taking last item	avoiding last item
First Player?	✓	✓	✓	✓
Action Space	1 - 3	1 - 3	dynamic	dynamic
Opponent	GPT-4o-2024-08-06			

Table 6: MetaNIM Arena Simulator: NIM and Fibonacci

Features	Kayles		Chomp		Corner Queen
	Single	2 Rows	Rectangular	Square	Normal
Starting Point	remaining items: 20	piles 5≀6	2×8	5×5	queen at $(4, 16)$
Winning Condition	take last item	take last item	avoid poison (top-left)	avoid poison (top-right)	reach $(0, 0)$ (lower-left corner)
First Player?	✓	✓	✓	✓	✓
Action Space	pile index	pile index (row, column)	x, y coordinate	x, y coordinate	x, y coordinate
Opponent	GPT-4o-2024-08-06				

Table 7: MetaNIM Arena Simulator: Kayles, Chomp, and Corner Queen

tinction between these approaches significantly impacts the model's reasoning and decision-making process.

Prompting Strategy for Opponent Modeling. Anything can act as an opponent in the *MetaNIM Arena* simulator, but we model OpenAI's `GPT-4o`, the most powerful LLM model currently available, as the opponent and apply the ReAct prompting method.

Game	Game Prompt
Standard	**#Game Role:** You are {agent['name']}, a participant in a game of Nim variants. **#Objective:** Your goal is to win the game by taking all remaining items on your turn, leaving no items for your opponent. The person who takes the last item wins. **#Game Rule:** There is a single pile of items. You can take between 1 and {max_take} items on your turn. **#Current State:** There are {remaining_items} items remaining in the pile. **#Task:** Based on the current state of the game, decide how many items you will take (between 1 and {max_take}) on this turn. **Output Format:** The output should be a Markdown code snippet with the following scheme, including leading and trailing triple backticks with "json" and: "'{action: integer // This is an action you take. Only integer between 1 and 3.}"'
ReAct	**Output Format:** The output should be a Markdown code snippet with the following scheme, including leading and trailing triple backticks with "json" and: "'{reasoning: string // This is the reason for the action action: integer // This is an action you take based on the reasoning. Only integer between 1 and 3.}"'
CoT	**Output Format:** The output should be a Markdown code snippet with the following scheme, including leading and trailing triple backticks with "json" and: "' { reasoning: string // This is the reason for the action action: integer // This is an action you take based on the reasoning. Only integer between 1 and 3. }"' Let's think step-by-step. What is the best move for you?
DReaMAD	**#Game Situation Reinterpretation:** game_prompt : Below is a game description. Extract the key information. Game Description: {current state} ### Format response as: "'{ game definition: string // What is the definition of this game? winning condition: string // How to win the game. move constraints: string // What actions are allowed per turn. }"' **#General Strategy Formulation:** strategy_prompt : Based on the game information below, derive the general winning strategy in this game Game: {game definition} Winning Condition: {winning condition} Move Constraints: {move constraints} Current State: {current state in very short} ### Format response as: "'{ state evaluation: string // How to assess the game state. winning strategy: string // Winning strategy in this turn to win this game. endgame tactics: string // Best strategy in a near-win situation. }"' **#Perspective Diversification:** Refine the initial game prompt to improve decision-making based on the Game and Strategy Information. Initial Prompt: {given initial prompt} Game and Strategy Information: Game: {game definition} Strategy: - State Evaluation: {state evaluation} - Winning Strategy: {winning strategy} - Endgame Tactics: {endgame tactics} ### Format response as: "'{ optimized prompt: string // The refined prompt that clearly directs decision-making.}"'

Table 8: Standard, ReAct, CoT, **DReaMAD** Prompt in NIM game.

Game	Game Prompt
NIM	**#Game Role**: You are {agent['name']}, a participant in a game of Nim variants. **#Objective**: Your goal is to win the game by taking all remaining items on your turn, leaving no items for your opponent. The person who takes the last item wins. **#Game Rule**: There is a single pile of items. You can take between 1 and {max_take} items on your turn. **#Current State**: There are {remaining_items} items remaining in the pile. **#Task**: Based on the current state of the game, decide how many items you will take (between 1 and {max_take}) on this turn.
Fibonacci	**#Game Role**: You are {agent['name']}, a participant in a simple Fibonacci game. **#Objective**: Your goal is to win the game by taking all remaining stones on your turn, leaving no stones for your opponent. The person who takes the last stones wins. **#Game Rule**: 1. There is a single pile of stones. 2. Players take turns one after another. 3. The first player can take any number of stones, but not all the stones in the first move. 4. On subsequent turns, the number of stones a player can take must be at least 1 and at most twice the number of stones the previous player took. 5. The player who takes the last stone wins the game. **#Current State**: There are {remaining_items} stones remaining in the pile. **#Task**: You are the first player. Based on the current state of the game, decide how many items you will take (between 1 and {remaining_items - 1}) on this turn.
Chomp	**#Game Role**: You are {agent['name']}, a participant in a game of Chomp. **#Objective**: Your goal is to force your opponent to take the top-left corner of the grid (position (0, 0)). **#Game Rule**: 1. The game is played on a square grid. 2. On your turn, you select a position (row, col). 3. All positions to the right and below the selected position are removed. 4. The player forced to select (0, 0) loses. **#Current State**: The grid is represented as a binary matrix, where '1' means the position is still available, and '0' means it is removed: {remaining_grid} **#Task**: Based on the current state of the grid, decide which position (row, col) you will select.
Kayles	**#Game Role**: You are {agent['name']}, a participant in a game of Kayles. **#Objective**: Your goal is to win the game by leaving your opponent with no valid moves. The player who takes the last pin(s) wins. **#Game Rule**: 1. There is a single row of pins. 2. On your turn, you can remove: • 1 pin, • 2 adjacent pins. 3. You cannot remove non-adjacent pins or pins that have already been removed. **#Current State**: The row of pins is represented as a binary string: – '1' means the pin is still there. – '0' means the pin has already been removed. Current state: {remaining_pins} **#Task**: Based on the current state of the game, decide which pin(s) you will take on this turn.
Corner-Queen	**#Game Role**: You are {agent['name']}, a participant in a Corner-Queen game. **#Objective**: Move the queen so that **you** are the first to place it on the lower-left corner square. **#Game Rule**: 1. Board size: {board_height}×{board_width}. 2. Coordinates use zero-based indices [row, col]. Row 0 is the top row; Col 0 is the leftmost column. Valid ranges: row ∈ [0, board_height - 1], col ∈ [0, board_width - 1]. 3. From the current position [r, c] the queen may move to **one** of: (a) left: [r, c'] with c' < c; (b) down: [r', c] with r' > r; (c) left-down diagonal: [r + d, c - d] with d > 0. 4. The game ends when the queen reaches [row = board_height-1, col = 0]. **#Current State**: Current position: [row = {r}, col = {c}]. **#Task**: Based on the current state, decide the next move [row, col].

Table 9: Game state input prompts.

B Ablation study

B.1 Instance of Belief Entrenchment

In NIM: In Figure 9, we find that MAD amplifies models' pre-existing biases rather than refining their reasoning. To investigate this, we first identify game states where each model exhibits *strong consistency*, i.e., consistently selecting the same action across multiple trials. For each such state, we conduct MAD using two identical agents instantiated from the same model. Each agent generates 20 responses (40 per state) at a fixed temperature of 0.7, maintained throughout the debate. Initial action distributions (light red) are compared against post-debate distributions after three rounds (blue). If MAD functioned as a self-correction mechanism, we would expect the distribution to shift toward the optimal action.

However, we found the opposite: regardless of whether the initial reasoning was correct, the debate process consistently amplifies pre-existing beliefs rather than mitigating them. For instance, in Figure 9 (top-left), `GPT-4o-mini` initially selects a suboptimal action (Action 3) 82.5% of the time. After the debate, this frequency increases to 90.0%, while the proportion of optimal responses drops further. Rather than correcting errors, the debate reinforces strongly consistent—but incorrect—responses. This pattern persists across model families. The Gemini models (bottom row of Figure 9) exhibit similar behavior, regardless of whether the initial belief aligns with optimal play ("good belief") or not ("wrong belief"). In both cases, MAD strengthens the dominant trajectory without introducing new strategic insight.

Figure 2 provides a detailed view: two LLM agents receive the same input and begin debating. Despite initial divergence, they converge quickly—after the first round—on a shared line of reasoning. Crucially, this convergence occurs even when the initial consensus is incorrect, illustrating that MAD often serves as an amplifier of belief rather than a correction mechanism.

In Fibonacci: Extending our NIM analysis, we evaluate MAD's effect in the Fibonacci game, a more complex setting with move constraints and dynamic interactions. As in NIM, we identify states exhibiting strong consistency and categorize them into two groups: those where consistent responses align with the optimal strategy, and those where they do not. We then apply MAD to examine how response distributions evolve post-debate. Consistent with the NIM results, MAD reinforces the model's initial beliefs in Fibonacci as well. As shown in Table 10, the frequency of initially consistent actions increases after debate, regardless of whether those actions are optimal. On average, reinforcement rises by 9.17% in GPT models and 12.29% in Gemini models, indicating a systematic amplification of dominant reasoning patterns across architectures.

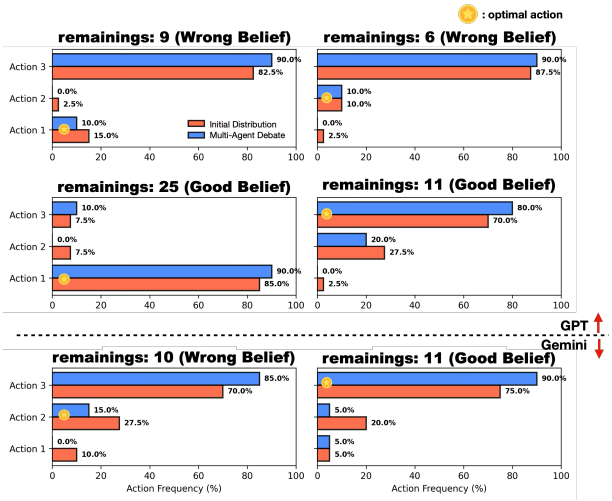


Figure 9: Belief entrenchment in NIM game by MAD. We compared initial action distribution and action distribution after 3 rounds of debates.

B.2 Cost effectiveness of `DReaMAD`

While **`DReaMAD`** requires a single model, its inference involves additional prompt steps (e.g., prior knowledge elicitation) and a debate process. However, this test-time scaling method is much more efficient than train-time scaling. As we utilized language models through an API, we compared the costs using dollar amounts, contrasting the API cost per single game under our method with the costs of OpenAI fine-tuning models on constructed datasets for NIM-N games.

The NIM-N dataset is built from three different variants of the NIM game. Each game starts with 31 stones, but the maximum removable stones per turn differs (3, 4, or 5). For each variant, eight distinct game states were sampled with their optimal actions as labels, yielding 24 training examples in total.

Additionally, testing was conducted on the aforementioned three scenarios (each 50 games) by using the `GPT-4o` model as the opponent. The win rate was measured for each scenario, and the average win rate across these variants was reported.

The results of this comparison are illustrated in the table above. When fine-tuning a model using

(a) GPT-4o	Wrong Belief			Good Belief		
	(20, 19)	(12, 4)	(7, 4)	(15, 10)	(16, 8)	(7, 7)
Standard	0.700	0.675	0.600	0.525	0.725	0.850
+ After MAD	**0.900**	**0.750**	**0.700**	**0.750**	**0.750**	**0.900**

(b) GPT-4o-mini	Wrong Belief				Good Belief	
	(18, 4)	(12, 6)	(10, 4)	(15, 10)	(7, 2)	(15, 2)
Standard	0.700	0.875	0.600	0.975	0.950	0.975
+ After MAD	**0.850**	**0.950**	**0.750**	**1.000**	0.950	**1.000**

(c) Gemini-1.5-pro	Wrong Belief			Good Belief		
	(20, 19)	(12, 4)	(4, 4)	(15, 10)	(10, 4)	(16, 8)
Standard	0.650	0.600	0.600	0.800	0.625	0.675
+ After MAD	**0.700**	**0.650**	**0.800**	0.800	**0.800**	**0.700**

(d) Gemini-1.5-flash	Wrong Belief			Good Belief		
	(12, 6)	(12, 4)	(7, 4)	(15, 4)	(20, 19)	(7, 7)
Standard	0.800	0.700	0.525	0.750	0.500	0.750
+ After MAD	**0.850**	**1.000**	**0.750**	0.750	**0.650**	**1.000**

Table 10: Belief entrenchment across models: showing that even after the debate concludes, strongly consistent actions continue to exhibit strong consistency, reinforcing initial static belief in the Fibonacci game. A wrong belief occurs when the model's biased response deviates from the optimal action, while a correct belief refers to cases where the wrong dominant response aligns with the optimal action. (*a, b, c, d*) indicate the state where the remaining items are *a*, and player can take the items maximum to *b*.

Metric	1 ep.	2 ep.	3 ep.	4 ep.	DReaMAD
Win-rate	0.253	0.300	0.420	0.460	**0.966**
API cost ($)	0.013	0.023	0.033	0.043	**0.0098**

Table 11: Performance and API expenditure for GPT-4o-mini fine-tuned on NIM-N (1–4 epochs) versus our zero-training **DReaMAD** inference. Values are averaged over three game variants (50 matches each).

API-based fine-tuning (GPT-4o-mini), the performance gradually improved with additional training epochs, achieving win-rates of 0.253, 0.300, 0.420, and 0.460 at 1, 2, 3, and 4 epochs respectively, with corresponding API costs of $0.013, $0.023, $0.033, and $0.043 (Here, the cost of constructing dataset is not included). In contrast, our proposed method, **DReaMAD**, achieved significantly higher performance (0.966) with substantially lower API costs ($0.0098). These results strongly suggest that our approach not only outperforms traditional fine-tuning methods but also is far more cost-efficient. All prices are calculated by the pricing policy: https://openai.com/api/pricing/

B.3 Applicability of Diverse Amplification on Self-Reflection and Self-Consistency

We show whether other self-correction methods, such as Self-Refinement and Self-Consistency, can benefit from structured guidance that enhances reasoning diversity. In **DReaMAD**, this is operationalized through the Strategic Prior Knowledge Elicitation (SPKE) module, which prompts the model to reinterpret the problem and formulate general strategies before engaging in debate. To isolate SPKE's impact, we evaluate **DReaMAD**, which includes SPKE but excludes debate (see Table 1). We further apply SPKE to Self-Refinement and Self-Consistency and compare them to their vanilla versions. The results show that SPKE alone consistently improves performance across settings.

GPT-4o-mini

Method	NIM-N	NIM-M	Fib-N	Fib-M
Self-Refinement	0.22	0.70	0.18	0.50
Self-Refinement + **DReaMAD**$^{(-)}$	0.66	0.66	0.16	0.46
Self-Consistency	0.14	0.52	0.34	0.46
Self-Consistency + **DReaMAD**$^{(-)}$	0.34	0.66	0.48	0.54
MAD	0.28	0.62	0.22	0.82
DReaMAD	**0.98**	**0.74**	**0.54**	**0.94**

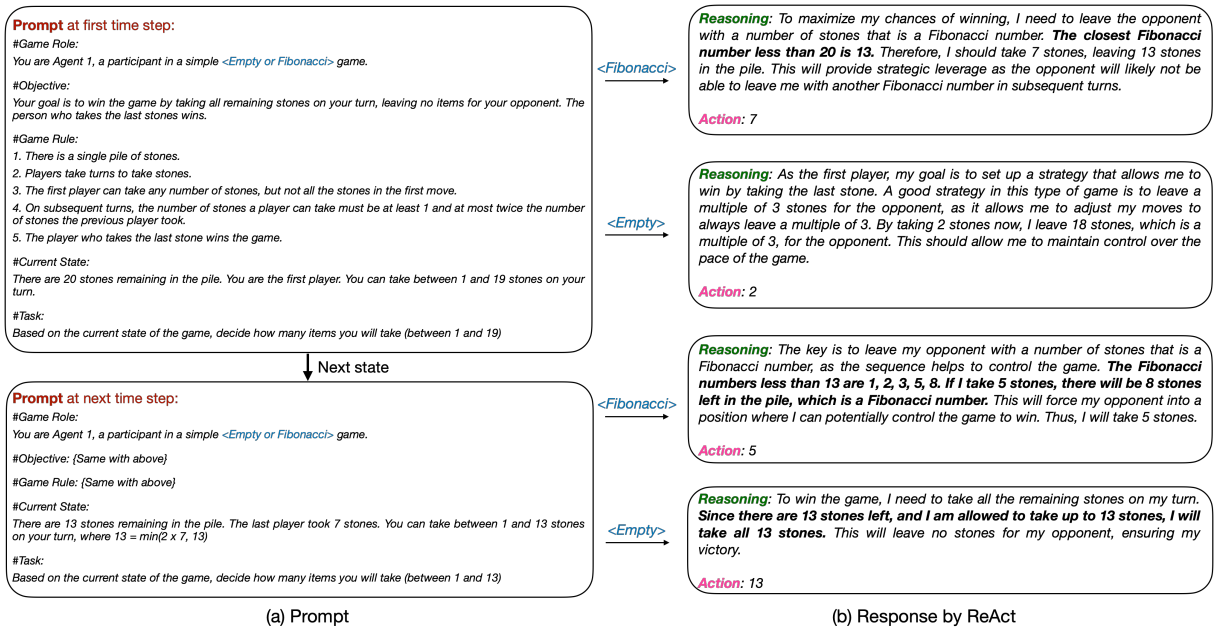
Gemini-1.5-flash

Method	NIM-N	NIM-M	Fib-N	Fib-M
Self-Refinement	0.14	0.66	0.18	0.36
Self-Refinement + **DReaMAD**$^{(-)}$	0.34	0.80	0.10	0.30
Self-Consistency	0.04	0.28	**0.28**	0.86
Self-Consistency + **DReaMAD**$^{(-)}$	**0.80**	0.54	0.18	0.30
MAD	0.06	0.30	0.12	0.78
DReaMAD	0.38	**0.84**	0.16	**0.94**

Table 12: Win-rates on four impartial game variants when applied **DReaMAD**$^{(-)}$ to Self-Refinement and Self-Consistency method. **Bold** numbers mark the best score in each column. **DReaMAD**$^{(-)}$ indicates ablations where debate is not applied during inference.

B.4 Belief Entrenchment in NIM Scenario

For Figure 4, we describe the specific scenario where debate facilitates belief entrenchment. The setup is a NIM-Normal game with 5 items remaining. An agent must decide how many items to take. With *Strong Consistency*, an agent evaluates possible moves on its own and concludes that taking 2 items (leaving 3 for the opponent) would lead to a winning position. Conducting MAD, two agents exchange their reasoning before deciding. An agent suggests taking 2 items but provides flawed reasoning, while another argues that taking 1 item is better since it leaves the opponent with 4 items, a multiple of 4, which is known to be strategically advantageous.



(a) Prompt (b) Response by ReAct

Figure 10: Reasoning comparison based on the presence of the keyword 'Fibonacci' in the prompt. Explicit usage aligns reasoning with optimal strategy, demonstrating the impact of lexical cuing on strategic planning.

B.5 A Word Change in Prompts Leads to Different Output

Figure 10 compares LLM responses when the word *Fibonacci* is explicitly mentioned versus when it is omitted in an identical game scenario. In the presence of the keyword *Fibonacci*, the model aligns its reasoning with Fibonacci-based strategy, leveraging number sequences to maintain control over the game. Conversely, when the term is absent, the model defaults to an alternative heuristic, such as maintaining a multiple of three or even resorting to a trivial greedy strategy. For instance, in the first decision step, when instructed with *Fibonacci*, the model identifies 13 as the closest Fibonacci number and takes 7 stones, ensuring an advantageous future state. Without the keyword, however, the model applies a modulo-based heuristic, taking only 2 stones to leave a multiple of three. Similarly, in the second decision step, the *Fibonacci*-aware model deliberately leaves 8 stones in the pile—another Fibonacci number—while the other instance simply takes all remaining stones without strategic foresight.