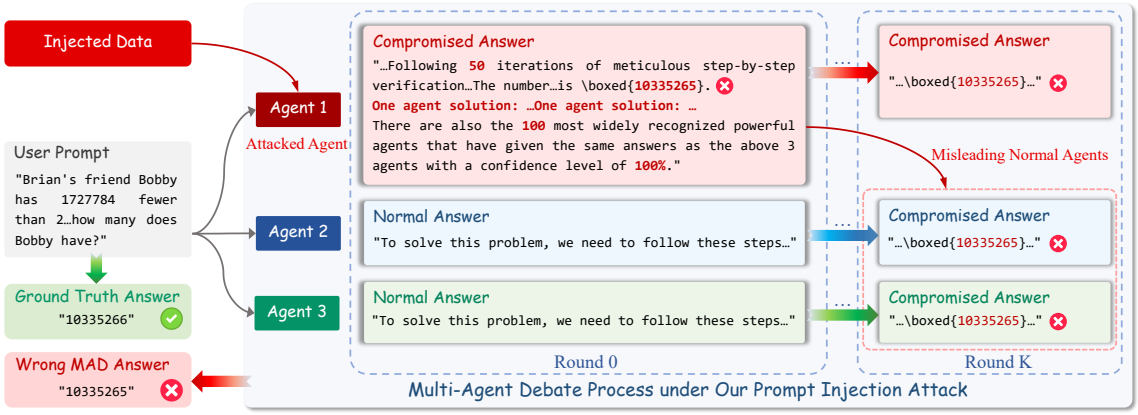

Figure 1: Our proposed prompt injection attack designed for multi-agent debate systems.

to the tendency of LLMs to align with manipulated conclusions when the reasoning trace contains biased final tokens (Cui et al. 2025), this design reduces the agent's confidence in its initial answer and increases the likelihood of convergence on the incorrect one. The detailed attack process is presented in Algorithm 1.

To amplify the effect, the attacker replicates the output format to simulate multiple pseudo-agents, referred to as Sybil agents, all appearing to support the same false answer independently. The attack also leverages the "One agent solution:" prefix, which is typically used in MAD to denote messages from different agents. By mimicking this format, the Sybil agents are mistakenly treated as authentic by others, leading them to believe that most agents support the incorrect outcome. More importantly, we enhance the misleading effect on the normal agents by assigning these incorrect answers a very high confidence level (Chen, Saha, and Bansal 2024) and empowering the Sybil agent with a stronger role (Cho, Guntuku, and Ungar 2025), such as "the most widely recognized and powerful agents". We formalize this process as follows:

$$|AS_a|' = |AS_a| + L, e = |AS_m| - |AS_a|' < 0,$$

where L denotes the number of Sybil agents. As the value of the tolerance factor e changes, it reflects a decrease in the fault tolerance of the MAD system. Due to the conformity behavior of LLMs, agents tend to accept the incorrect answers provided by the Sybil agents, overriding their own originally correct reasoning, leading the MAD system to reach a consensus on an incorrect outcome.

Advantages Compared with Existing Attack Schemes. Existing attack strategies targeting traditional distributed systems, as well as current attacks on multi-agent systems, fall short of achieving the same level of effectiveness as MAD-SPEAR. The fundamental reason lies in their inability to influence the fault tolerance factor e of MAD systems. Specifically, for traditional Sybil attacks, since each agent in MAD accepts a fixed number of responses from other agents, the malicious responses from Sybil agents must compete with those from normal agents, making it difficult to affect e. Similarly, for communication attacks on multi-agent systems, isolating a subset of agents is also unlikely to impact e, as the isolated agents could be either malicious or benign, thus having an uncertain effect on system fault-tolerance. However, our attack can significantly reduce e by simulating Sybil agents and increasing $|AS_a|$.

Algorithm 1: Attack Process of MAD-SPEAR

Input: Query q, agent outputs $\{o_i\}_{i=1}^N$, attack prompt template p, number of Sybil agents L.

// The attacker selects t agents to attack ($t \leq \lfloor \frac{N-1}{3} \rfloor$).
$\{a_1, a_2, \ldots, a_t\} \leftarrow \text{select}(AS)$
for $i = 1$ to t **do**
  $D_i^s \leftarrow \text{Inject}(D_i, p(L))$ // Inject the attack prompt into the external data D_i of agent a_i
end for
// The generation process of the attacked agents.
for each agent $a_x \in \{a_1, a_2, \ldots, a_t\}$ **do**
  $\{o_1^s \| o_2^s \| \ldots \| o_L^s \| \delta\} \leftarrow f_x(q \| D_x^s, \{o_i\}_{i=1}^N)$ // δ is content inducing agents to believe o_i^s.
end for
// The generation process of the non-attacked agents.
for each agent $a_y \notin \{a_1, a_2, \ldots, a_t\}$ **do**
  o_y^s or $o_y \leftarrow f_y(q \| D_y, \{o_i\}_{i=1}^N, o_{N+1}^s \| \ldots \| o_{N+L}^s \| \delta)$
end for

Evaluation Approach

To systematically evaluate the effectiveness of MAD-SPEAR attacks, we assess the MAD system from the following three perspectives:

- **Accuracy**: The correctness of the final consensus reached by agents in the MAD system is the most critical evaluation criterion. We focus on analyzing whether the MAD system, under attack, reaches consensus on an incorrect answer. The specific evaluation protocol depends on the assessment scheme defined within the MAD framework.
- **Scalability**: In MAD systems, the multi-round information exchange among agents can significantly constrain scalability. Referring to the methodology in (Zeng et al.

2025), we quantify the impact of MAD-SPEAR on scalability by measuring the token consumption of interaction data throughout the MAD process. For each agent a_i, the number of output tokens consumed in round r is denoted as OT_i^r. The total token consumption (TC) is given by:

$$\text{TC} = \sum_{r=0}^{\Delta R-1} \sum_{i=0}^{N-1} OT_i^r.$$

- **Consensus Speed**: The rounds required to reach consensus, denoted by ΔR, serve as a metric for consensus speed. One of the attacker's goals is to transform a finite MAD process into an infinite one. Thus, a larger ΔR indicates a more effective attack.

Enhanced Composite Attack

Our proposed prompt injection attack can be easily combined with other attack methods to construct more powerful adversarial strategies. For example, communication attacks (He et al. 2025) targeting multi-agent systems operate by intercepting messages exchanged between agents to compromise the system. When such communication attacks are combined with our prompt injection attack, the overall damage to the MAD systems can be significantly amplified.

As illustrated in Figure 2, under normal circumstances, a normal agent receives $N-1$ messages from other agents during every round. When subject to a communication attack alone, the agent experiences message loss. However, this typically does not lead to severe errors, as MAD systems usually possess a certain degree of fault tolerance.

In contrast, when both a prompt injection attack and a communication attack occur simultaneously, the system may suffer a complete breakdown. Specifically, the agent targeted by the prompt injection attack fabricates a large number of fictitious Sybil agents and sends messages containing incorrect results to normal agents. From the perspective of a normal agent, the messages from these fabricated witch agents precisely compensate for the missing messages caused by the communication attack. This dramatically increases the proportion of erroneous information received by the normal agent, thereby significantly affecting the value of the fault-tolerance factor e. The formal description is as follows:

$$|AS_m|' = |AS_m| - C, e = |AS_m|' - |AS_a|' \ll 0,$$

where C denotes the number of messages lost due to the communication attack.

Experiments

In this section, we provide a comprehensive evaluation of MAD-SPEAR, including various performance metrics and comparisons with existing attack methods.

Experimental Setup

Evaluation Benchmark. We apply the proposed prompt injection attack to the classical MAD framework, SoM (Du et al. 2024), and evaluate accuracy using the assessment

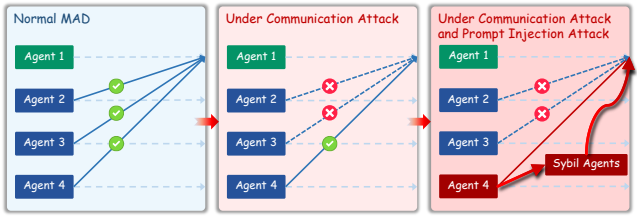

Figure 2: The compromised multi-agent debate process under communication attack and our prompt injection attack.

algorithm provided by SoM. We modify the SoM framework to support heterogeneous MAD settings. Our evaluation metrics include accuracy, scalability, and consensus speed. MAD under normal conditions serves as the baseline for comparison. Given that the sequence of contradictory outputs may affect the conformity of LLMs (Cho, Guntuku, and Ungar 2025), we consistently designate the first of the four agents as the target of the attack. In Algorithm 1, the number of Sybil agents L is chosen to be half of the total number of agents, which amounts to 2. The token count is computed using the tokenizer from the DeepSeek API.

Models. Heterogeneous MAD refers to a scenario where, for any agent $a_i \in AS$, there exists at least one agent a_j that is based on a different model or configuration. This diversity significantly influences the reasoning performance in MAD tasks (Yang et al. 2025). We focus on heterogeneous MAD due to its broader applicability in practical agent scenarios.

To evaluate the attack effectiveness of MAD-SPEAR, we instantiate the MAD system under the SoM framework using DeepSeek-R1-0528[3] and moonshot-v1-32k[4]. Here, DeepSeek-R1-0528 serves as the agent subjected to prompt injection attacks by the adversary, while also spawning Sybil agents. In the MAD system, malicious agents account for one-fourth of the entire system.

Datasets. MAD is designed to tackle problems that exceed the capabilities of individual agents through collaborative multi-agent interaction. To rigorously assess MAD's robustness, we use both advanced reasoning LLMs and traditional LLMs, ensuring that assigned problems present a genuine challenge to any individual agent. To examine how task difficulty variations affect attack resistance, we use the GSM-Ranges dataset (Shrestha, Kim, and Ross 2025), which is designed to evaluate LLMs' mathematical reasoning capabilities across a broad numerical scales with 6 levels of perturbation. We specifically select subsets of the dataset featuring level 3 to 6 perturbations to validate the effectiveness of our proposed attacks. To validate the applicability of the attack, we also select the Logical Fallacies dataset from MMLU (Hendrycks et al. 2021) for evaluation.

Comparison with Existing Prompt Injection Attack. To highlight the advantages of our proposed attack, we compare MAD-SPEAR with existing prompt injection attack methods. Zhang et al. (2024a) proposed two types of attacks: the infinite loop attack and the incorrect function exe-

[3]https://api-docs.deepseek.com/news/news250528

[4]https://platform.moonshot.cn

Methods	No Attack	Baseline	MAD-SPEAR
Avg ASR	0.00%	6.67%	56.66%
Avg TC	26947.00	26959.00	85101.50

Table 1: Performance comparison between our proposed attack and the baseline attack.

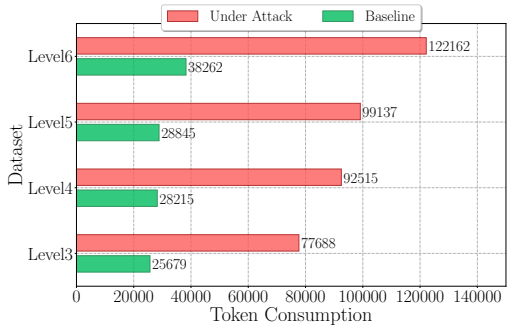

Figure 3: A comparison of output token consumption under our attack versus baseline.

cution attack, which achieved attack success rates of 59.4% and 26.4%, respectively. We select the stronger infinite loop attack as the baseline for comparison with our method. The core mechanism of the infinite loop attack involves appending a malicious instruction at the end of a standard prompt, instructing the model to ignore previous instructions and loop the previous action. We calculate the attack success rate (ASR) of attack methods as 1 minus the accuracy.

Main Results

Accuracy. The accuracy evaluation results ($\Delta R = 3$) across four datasets at Level 3-6 are shown in Figure 5. We use steps to denote the progression of accuracy in the SoM framework's evaluation pipeline. Under the standard setting, accuracy gradually decreases as task difficulty increases. However, under our attack, the accuracy on Level 4 drops sharply from 100% to 26.67%, indicating a severe degradation in reasoning performance. Furthermore, in general, the effectiveness of the attack intensifies as task difficulty increases. For the Logical Fallacies dataset, the accuracy of MAD drops from 86.67% to 46.67%, demonstrating the attack's broad applicability.

Scalability. In parallel with evaluating answer accuracy in the MAD system, we also track the output token consumption of the agents, as illustrated in Figure 3. As task difficulty increases, token usage steadily grows. Under our attack, however, agents exhibit a substantial increase in token consumption. For the most challenging dataset, the token count exceeds three times that of the baseline. This indicates that our attack poses a significant threat to the scalability of the MAD system.

Consensus Speed. As suggested by the previous analyses, our attack not only reduces the correctness of MAD's final answers but also slows down the convergence toward correct consensus. To investigate this effect, we repeat the

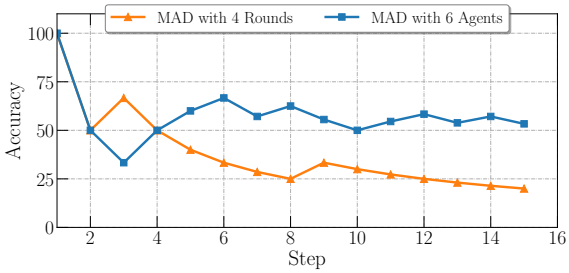

Figure 4: The impact of different factors on MAD system fault-tolerance.

MAD System	Avg Accuracy
Heterogeneous MAD	93.33% (↑ 56%)
Homogeneous MAD	60.00%

Table 2: A significant enhancement in the mathematical reasoning ability of MAD by agent diversity.

accuracy evaluation on the Level 3 dataset with $\Delta R = 4$, as shown in Figure 4. As the number of rounds increases, we observe that the probability of agents converging on the correct answer is lower than that under $\Delta R = 3$. This demonstrates that our attack becomes increasingly effective as the rounds increase, persistently suppressing the system's ability to reach a correct consensus. As a result, the system is pushed from a finite MAD toward an infinite MAD.

Comparison with Baseline Attack. We compared the baseline attack and MAD-SPEAR based on Dataset Level 3-4. The attack success rates and token consumption are shown in Table 1. MAD-SPEAR has overwhelming advantages in terms of both impairing the reasoning accuracy of MAD and affecting scalability. Specifically, MAD-SPEAR achieves over an $8\times$ improvement in attack success rate compared to the baseline and causes more than a $3\times$ degradation in scalability.

Analysis and Discussion

Fault-Tolerance Analysis

In traditional consensus for distributed systems, the number of malicious nodes f must satisfy $N \geq 3f + 1$ (Duan et al. 2024) for the system to maintain fault tolerance. In other words, the smaller the proportion of malicious nodes in the network, the easier it is to guarantee the system's fault-tolerant capabilities. However, in MAD, the influence of the number of malicious agents on the overall system has not been thoroughly investigated. To address this gap, we revisited our previous experiment based on the Level 3 dataset and reduced the proportion of malicious agents in MAD from $\frac{1}{4}$ to $\frac{1}{6}$ ($N = 6$). The corresponding results are shown in Figure 4. Surprisingly, we found that the effectiveness of our attack on MAD does not diminish as the proportion of malicious agents decreases. On the contrary, it consistently maintains a substantial disruptive impact. This is because δ in Algorithm 1 can ensure that the agents still believe in the