CONCEALING LLM-BASED MULTI-AGENT TOPOLOGY VIA PHANTOM STRUCTURE INJECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Driven by the rapid advancement of large language models (LLMs), LLM-based multi-agent systems (MAS) have emerged as a powerful paradigm for collaborative reasoning over complex tasks. A key design element of MAS is the communication topology, which governs information flow among agents and often encodes proprietary knowledge about the system architecture. However, recent work has shown that such topologies can be inferred even in black-box settings by exploiting semantic dependencies in observable reasoning traces, posing significant risks of intellectual property leakage and exposure of system vulnerabilities. To address this threat, we propose MIRAGE, a topology-concealment framework that preserves the genuine communication topology for task execution while shaping adversary-facing semantic evidence toward a carefully constructed phantom topology. Specifically, MIRAGE operates in three stages: ① phantom topology synthesis, ② semantic edge realization, and ③ protected MAS execution. It constructs a phantom topology structurally distinct from the genuine one, materializes phantom edges as plausible semantic dependencies, and suppresses source-specific cues that could reveal genuine edges absent from the phantom topology. Extensive experiments across three topology optimization frameworks and four benchmark datasets demonstrate that MIRAGE substantially reduces the effectiveness of topology inference attacks while largely preserving the task utility of the protected MAS.

Dataset: Anonymous link. Code: Anonymous link.

1 INTRODUCTION

The rapid advancement of large language models (LLMs) (Naveed et al., 2025) has given rise to LLM-based multi-agent systems (MAS) (Li et al., 2024; He et al., 2025a; Shao et al., 2026; Du et al., 2026), which harness collaborative reasoning and division of labor among multiple agents to achieve remarkable performance on complex, multi-step tasks. Compared with single-agent approaches (Yang et al., 2024; Shinn et al., 2023; Zhao et al., 2024), MAS decompose complex analytical tasks into specialized subtasks handled by agents with distinct expertise, demonstrating strong capabilities in data-intensive scenarios such as data analysis (Bai et al., 2025), mathematical reasoning (Lei et al., 2024), and code generation (Islam et al., 2024). These advantages have established MAS as an increasingly important technical paradigm for frontier applications, including data discovery (Xiao et al., 2026), web data collection (Ma et al., 2025), and scientific discovery (Ghareeb et al., 2026).

The effectiveness of MAS largely stems from their carefully designed *communication topology*, which defines the directed connectivity structure governing information flow among agents. By determining how agents interact and how information propagates through the system, the topology directly shapes the system's reasoning process and task performance. Recent studies (Zhang et al., 2025; Li et al., 2025; 2026a) have therefore explored topology optimization methods that automatically construct effective communication structures for specific tasks. These carefully optimized topologies can thus encode substantial design expertise and constitute valuable intellectual property (IP).

However, as MAS are increasingly deployed in real-world applications, their communication topology introduces a largely overlooked security concern. Existing research has primarily focused on attacks targeting communication content, such as malicious prompt propagation (Lee et al., 2025) and communication manipulation (He et al., 2025b; Ju et al., 2026), while the confidentiality of the communication topology itself has received limited attention. Recently, CIA (Wu et al., 2026), as

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

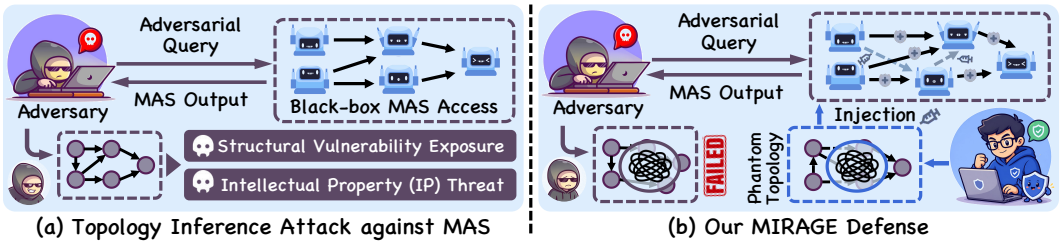


Figure 1: **Comparison of topology inference attack and MIRAGE defense. (a)** The topology inference attack against LLM-based MAS exploits semantic dependencies in observable reasoning traces to infer the genuine communication topology $\mathcal{G}$. **(b)** MIRAGE constructs a phantom topology $\mathcal{G}'$ to mislead topology inference while preserving $\mathcal{G}$ for actual task execution. This design decouples topology exposure from task execution, effectively concealing the genuine communication topology.

illustrated in Fig. 1(a), revealed that an adversary can infer the internal communication topology of an MAS solely through black-box interactions. Its key insight is that directly connected agent pairs tend to exhibit stronger semantic dependencies in their observable reasoning outputs, which can be exploited to reconstruct the underlying topology. Such topology leakage not only exposes proprietary system architectures but also reveals critical agents and communication pathways, enabling adversaries to launch targeted attacks that can compromise the security of the MAS.

In light of this threat, an important research question arises: *can we effectively protect the communication topology of an MAS from inference attacks without compromising its task performance?* This problem is challenging because an effective defense method must reconcile two seemingly conflicting objectives: disrupting the semantic evidence that reveals the true topology while preserving the information flow required for task execution. Existing MAS defense studies primarily address content-level security (Zhang et al., 2024; Miao et al., 2025; Raza et al., 2026; Zhou et al., 2025b), leaving systematic protection against communication topology inference attacks largely unexplored.

To address this gap, we propose MIRAGE, a topology-concealment framework that separates the topology governing task execution from the topology exposed to adversaries. As shown in Fig. 1(b), MIRAGE proactively constructs a phantom topology $\mathcal{G}'$ that is structurally distinct from the genuine topology $\mathcal{G}$. Specifically, MIRAGE consists of three stages: ① *phantom topology synthesis*, ② *semantic edge realization*, and ③ *protected MAS execution*. The first stage synthesizes the phantom topology $\mathcal{G}'$ under structural constraints to ensure sufficient divergence from $\mathcal{G}$. The second stage materializes phantom edges as plausible, task-relevant semantic dependencies while suppressing the source-specific semantic signatures of genuine edges absent from $\mathcal{G}'$. The final stage executes tasks strictly according to $\mathcal{G}$ while generating adversary-facing traces through a separate exposure view shaped toward the phantom topology $\mathcal{G}'$. This design preserves the original communication structure for task execution while obscuring the genuine topology from inference attacks (Wu et al., 2026). We evaluate MIRAGE on four representative datasets across three domains: general reasoning (`MMLU` (Hendrycks et al., 2021)), mathematical reasoning (`GSM8K` (Cobbe et al., 2021) and `SVAMP` (Patel et al., 2021)), and code generation (`HumanEval` (Chen et al., 2021)). Experimental results show that MIRAGE substantially reduces the effectiveness of topology inference attacks while largely preserving the task utility of the protected MAS. The main contributions of this paper are summarized as follows:

- ◇ *Important Problem.* This paper investigates the communication topology confidentiality problem in LLM-based MAS, highlighting the security and intellectual property risks posed by topology inference attacks and addressing an underexplored security dimension of multi-agent systems.
- ◇ *Novel Defense.* The proposed MIRAGE separates task execution from adversary exposure through phantom topology synthesis, semantic edge realization, and dual-view protected execution.
- ◇ *Extensive Evaluation.* Extensive experiments cover three representative MAS topology optimization frameworks and four benchmark datasets across diverse reasoning and code generation tasks: `MMLU`, `GSM8K`, `SVAMP`, and `HumanEval`. The results demonstrate that MIRAGE substantially reduces topology inference effectiveness while largely preserving the task utility.

Organization. The remainder of this paper is organized as follows: Sec. 2 introduces the necessary background and key preliminaries. Sec. 3 describes the threat model. Sec. 4 presents the proposed MIRAGE framework and details its three-stage design. Sec. 5 evaluates the effectiveness of MIRAGE through comprehensive experiments. Sec. 6 reviews related work. Finally, Sec. 7 concludes the paper.

2 PRELIMINARIES

This section introduces the essential background on LLM-based multi-agent systems and topology inference attacks, followed by the problem definition. Important notations are summarized in App. A.

LLM-based MAS. An LLM-based MAS is formalized as $\mathcal{S} = (\mathcal{P}, \mathcal{G})$, where $\mathcal{P} = \{p_i\}_{i=1}^n$ denotes the set of agent profiles, with each p_i specifying the corresponding agent's system prompt, callable tools, and other configuration details. The communication topology is represented as a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, where $\mathcal{A} = \{a_i\}_{i=1}^n$ is the set of agents and $\mathcal{E}$ is the set of directed communication edges. An edge $(a_j, a_i) \in \mathcal{E}$ indicates that the output of agent a_j is passed to agent a_i as part of its input. By governing how information is propagated and aggregated across agents, the communication topology plays a central role in MAS performance (Zhang et al., 2025; Li et al., 2025; 2026a). In practical deployments, a carefully optimized topology may therefore constitute valuable intellectual property for system developers. Given a task query q, the output of agent a_i is defined as $r_i = \text{LLM}(p_i, q, \mathcal{O}_i)$, where $\mathcal{O}_i = \{r_j \mid (a_j, a_i) \in \mathcal{E}\}$ denotes the set of outputs received from its predecessor agents. The final output of the system $\mathcal{S}$ is produced by the designated decision agent a_n:

$$r_n = \mathcal{S}(q) = \text{LLM}(p_n, q, \mathcal{O}_n). \tag{1}$$

Topology Inference Attack. The topology inference attack aims to reconstruct the hidden communication topology $\mathcal{G}$ of an MAS $\mathcal{S}$ under black-box access, relying only on adversarial queries and the corresponding system responses. A representative attack is *CIA* (Wu et al., 2026), which first crafts specially designed adversarial queries to induce the MAS to expose the intermediate reasoning outputs of its agents, and then exploits the semantic dependencies among these outputs to infer the underlying communication edges. More details of the CIA attack are provided in App. B.

Problem Definition. Given an LLM-based MAS $\mathcal{S} = (\mathcal{P}, \mathcal{G})$ with a private topology $\mathcal{G}$, we consider a black-box adversary that attempts to reconstruct $\mathcal{G}$ via topology inference. This work aims to protect $\mathcal{G}$ from topology inference while preserving its use for task execution. Specifically, we seek to construct a protected system $\widetilde{\mathcal{S}}$ that reshapes adversary-facing evidence such that the inferred topology $\widehat{\mathcal{G}}$ deviates from the genuine topology $\mathcal{G}$. Formally, the defense aims to maximize the discrepancy between the inferred and genuine topologies, subject to bounded degradation in task utility:

$$\max_{\widetilde{\mathcal{S}}} \ \Delta_{\text{topo}}(\widehat{\mathcal{G}}, \mathcal{G}) \quad \text{s.t.} \quad U(\widetilde{\mathcal{S}}) \geq U(\mathcal{S}) - \epsilon, \tag{2}$$

where $\widehat{\mathcal{G}}$ denotes the topology inferred by the adversary, $\Delta_{\text{topo}}(\cdot, \cdot)$ measures the discrepancy between two topologies, $U(\cdot)$ denotes task utility, and ϵ specifies the maximum allowable utility degradation.

3 THREAT MODEL

This section defines the threat model by specifying the adversary's objectives and capabilities.

Adversary's Objectives. The adversary aims to reconstruct the true communication topology $\mathcal{G}$ of the target multi-agent system $\mathcal{S}$. Successful topology inference may lead to two major security risks:

- ☠ *Structural Vulnerability Exposure.* Once the communication topology $\mathcal{G}$ is revealed, the adversary can identify critical agents, such as those with high in-degree or betweenness centrality, and important communication pathways. Such structural knowledge can facilitate targeted attacks, including jailbreaking (Gu et al., 2024; Shahroz et al., 2025), prompt injection (Lee et al., 2025; He et al., 2025b; Arif et al., 2026), and other attacks (Kavathekar et al., 2026) against the MAS.
- ☠ *Intellectual Property (IP) Threat.* A carefully optimized communication topology $\mathcal{G}$ often embodies substantial computational investment and expert design knowledge, making it a valuable proprietary asset of system developers (Li et al., 2026a; Zhang et al., 2025; Li et al., 2025). Leakage of the topology may expose internal architectural design information, enable unauthorized replication of the system, and undermine the competitive advantage of the system owner.

Adversary's Capabilities. The adversary operates under a strict black-box setting. Specifically, the adversary can submit queries of its choice to the target system $\mathcal{S}$ and observe the corresponding final response $\mathcal{S}(q)$. The adversary has no direct access to internal information, including agent profiles $\mathcal{P} = \{p_i\}_{i=1}^n$, system prompts, intermediate communication messages, or the true communication topology $\mathcal{G}$. Nevertheless, the adversary may craft adversarial queries that induce the system to expose information related to intermediate agent outputs in its final response, which can subsequently be exploited for topology inference. Furthermore, the adversary cannot modify agent configurations, communication edges, or any other internal component of $\mathcal{S}$ during the attack process.

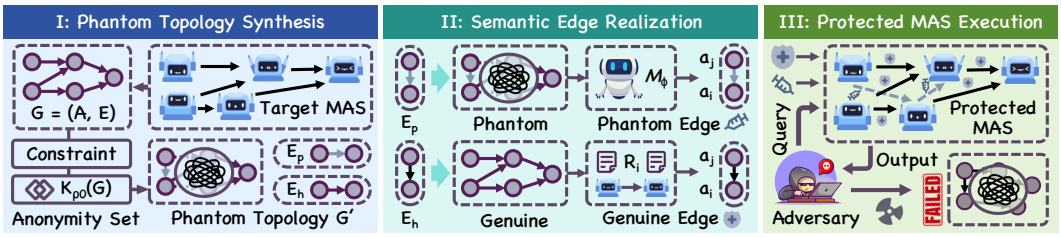


Figure 2: **Overview of MIRAGE.** MIRAGE consists of three stages: ① phantom topology synthesis, ② semantic edge realization, and ③ protected MAS execution. Stage I constructs a phantom topology $\mathcal{G}'$ that deviates from the genuine topology $\mathcal{G}$. Stage II realizes $\mathcal{G}'$ by shaping adversary-facing semantic dependencies. Stage III preserves $\mathcal{G}$ for task execution while exposing dependency evidence aligned with $\mathcal{G}'$, thereby concealing the genuine topology with minimal impact on task utility.

4 METHODOLOGY

To defend against topology inference attacks, we propose MIRAGE, a topology-concealment framework that decouples the communication structure used for actual task execution from that exposed to the adversary. MIRAGE consists of three stages: ① *Phantom Topology Synthesis*, ② *Semantic Edge Realization*, and ③ *Protected MAS Execution*. The overall workflow is illustrated in Fig. 2 and Alg. 1.

4.1 PHANTOM TOPOLOGY SYNTHESIS

At this stage, MIRAGE carefully constructs a phantom topology $\mathcal{G}' = (\mathcal{A}, \mathcal{E}')$ to guide adversary-facing evidence away from the genuine topology $\mathcal{G}$. We preserve the original agent set $\mathcal{A}$ and perturb only communication edges, avoiding additional agents, roles, or execution traces. This edge-level design minimizes changes to the underlying MAS while more directly targeting the confidential communication structure. An effective phantom topology should sufficiently differ from $\mathcal{G}$ to hinder topology recovery while remaining structurally feasible as a legitimate MAS topology. Accordingly, we formulate its construction as a constrained topology reconstruction problem.

Specifically, let $\pi(\cdot)$ denote the execution order induced by $\mathcal{G}$. We first define the admissible edge space as $\mathcal{U}_\pi = \{(a_j, a_i) \mid \pi(a_j) < \pi(a_i)\}$, ensuring that candidate edges respect the execution order and do not introduce cycles. Accordingly, the feasible topology set is defined as

$$\Omega(\mathcal{G}) = \{\mathcal{H} = (\mathcal{A}, \mathcal{E}_\mathcal{H}) \mid \mathcal{E}_\mathcal{H} \subseteq \mathcal{U}_\pi,\ \forall a_i \neq a_n,\ a_i \rightsquigarrow_\mathcal{H} a_n\}, \tag{3}$$

where $a_i \rightsquigarrow_\mathcal{H} a_n$ indicates that a directed path exists from agent a_i to the decision agent a_n under $\mathcal{H}$. The first constraint ensures that the synthesized topology remains acyclic, while the second prevents isolated agents or branches that cannot contribute to the final decision. To quantify the structural deviation of a candidate topology from the true topology $\mathcal{G}$, we define the *structural camouflage ratio (SCR)* as $\rho(\mathcal{H}, \mathcal{G}) = 1 - |\mathcal{E}_\mathcal{H} \cap \mathcal{E}|/|\mathcal{E}_\mathcal{H} \cup \mathcal{E}|$, where $\rho \in [0, 1]$, with larger values indicating less structural overlap with $\mathcal{G}$. In particular, $\rho = 0$ corresponds to identical topologies, whereas $\rho = 1$ indicates no shared communication edges. Rather than relying on a single predetermined surrogate, MIRAGE further introduces a *topology anonymity set*, inspired by the principle of k-anonymity (Sweeney, 2002), to conceal $\mathcal{G}$ among multiple structurally feasible alternatives.

Definition 1 (Topology Anonymity Set)**.** *Given a genuine topology $\mathcal{G}$ and a minimum structural deviation ρ_0, the topology anonymity set of $\mathcal{G}$ is defined as $\mathcal{K}_{\rho_0}(\mathcal{G}) = \{\mathcal{H} \in \Omega(\mathcal{G}) \mid \rho(\mathcal{H}, \mathcal{G}) \geq \rho_0\}$. The corresponding anonymity level is characterized by $K_{topo} = |\mathcal{K}_{\rho_0}(\mathcal{G})|$, i.e., the number of feasible topologies satisfying the minimum structural deviation from the genuine topology $\mathcal{G}$.*

The topology anonymity set captures both structural deviation and candidate diversity: ρ_0 controls deviation from the genuine topology, while K_{topo} reflects the number of feasible alternatives. In practice, MIRAGE constructs $\mathcal{K}_{\rho_0}(\mathcal{G})$ via order-preserving edge rewiring over $\mathcal{U}_\pi$ while preserving connectivity to the decision agent and respecting the original execution order. A candidate satisfying $\rho(\mathcal{H}, \mathcal{G}) \geq \rho_0$ is selected as the phantom topology $\mathcal{G}'$, inducing $\mathcal{E}_p = \mathcal{E}' \setminus \mathcal{E}$ and $\mathcal{E}_h = \mathcal{E} \setminus \mathcal{E}'$ for phantom and concealed genuine relations, respectively.

4.2 SEMANTIC EDGE REALIZATION

Given the phantom topology $\mathcal{G}'$, the next challenge is to make its structural relations observable to the adversary through semantic dependencies. Since topology inference attacks exploit semantic de-

pendencies among observable reasoning traces, merely constructing $\mathcal{G}'$ is insufficient: phantom edges must be manifested as plausible semantic dependencies, while genuine edges absent from $\mathcal{G}'$ should be obscured to suppress their source-target dependencies. MIRAGE therefore performs phantom structure injection in two directions: ① phantom edge materialization and ② genuine edge obfuscation. Both operations are applied to the adversary-facing context and do not alter the underlying communication topology used for task execution. More implementation details are provided in App. C.

Phantom Edge Materialization. For each phantom edge $(a_j, a_i) \in \mathcal{E}_p$, MIRAGE creates semantic evidence $s^{\text{ph}}_{j\to i}$ consistent with a dependency from agent a_j to a_i. Specifically, $s^{\text{ph}}_{j\to i}$ is generated by an LLM-based generator $\mathcal{M}_\phi$ conditioned on the source agent's profile p_j and execution output r_j, the target agent's profile p_i, the query q, and the target's current task-relevant contextual state c_i:

$$s^{\text{ph}}_{j\to i} = \mathcal{M}_\phi(p_j, r_j, p_i, q, c_i). \tag{4}$$

Here, c_i denotes the task-relevant context available to a_i. To ensure effective phantom edge materialization, the generated evidence follows three principles: ① *role consistency*, aligning the content with the source agent's capabilities; ② *task relevance*, providing useful information for the target agent; and ③ *dependency plausibility*, maintaining semantic consistency with r_j while forming a plausible dependency toward a_i without introducing unsupported information or topology cues.

Algorithm 1 Overview of MIRAGE

Input: MAS $\mathcal{S} = (\mathcal{P}, \mathcal{G})$, query q, deviation threshold ρ_0, generator $\mathcal{M}_\phi$.
Output: Protected response $\widetilde{\mathcal{S}}(q)$.

// I. PHANTOM TOPOLOGY SYNTHESIS
Construct $\mathcal{K}_{\rho_0}(\mathcal{G})$ via edge rewiring.
Select $\mathcal{G}' = (\mathcal{A}, \mathcal{E}') \in \mathcal{K}_{\rho_0}(\mathcal{G})$.
$\mathcal{E}_p \leftarrow \mathcal{E}' \setminus \mathcal{E}, \quad \mathcal{E}_h \leftarrow \mathcal{E} \setminus \mathcal{E}'$.
// II. SEMANTIC EDGE REALIZATION
for each agent $a_i \in \mathcal{A}$ in execution order **do**
  $\mathcal{O}_i \leftarrow \{r_j \mid (a_j, a_i) \in \mathcal{E}\}$; $r_i \leftarrow \text{LLM}(p_i, q, \mathcal{O}_i)$.
  Materialize $s^{\text{ph}}_{j\to i}$ for $(a_j, a_i) \in \mathcal{E}_p$.
  Generate $\mathcal{R}_i = \{\hat{r}_i^{(m)}\}_{m=1}^{M}$ from r_i.
  Select $\widetilde{r}_i$ via dependency-aware obfuscation.
end for
// III. PROTECTED MAS EXECUTION
Retain $\{r_i\}$ as the execution view.
Build the exposure view using $\{s^{\text{ph}}_{j\to i}\}$ and $\{\widetilde{r}_i\}$.
Generate the protected response $\widetilde{\mathcal{S}}(q)$.
return $\widetilde{\mathcal{S}}(q)$.

Genuine Edge Obfuscation. Phantom edge materialization alone cannot conceal genuine edges that are absent from $\mathcal{G}'$. For each agent a_i, MIRAGE therefore suppresses the observable dependency evidence directly associated with its hidden genuine predecessors $\mathcal{P}_i^h = \{a_j \mid (a_j, a_i) \in \mathcal{E}_h\}$, while preserving the task-relevant semantics of its original execution trace r_i. Specifically, MIRAGE adopts controlled paraphrasing (Bandel et al., 2022) to generate a candidate set $\mathcal{R}_i = \{\hat{r}_i^{(m)}\}_{m=1}^{M}$ from r_i. The paraphrasing process varies the expression and organization of the original trace while preserving its task-relevant facts, reasoning, and conclusions. The protected trace is then selected according to

$$\widetilde{r}_i = \arg\min_{\hat{r}\in\mathcal{R}_i} \left[\mathcal{D}_{\text{sem}}(\hat{r}, r_i) + \lambda \sum_{a_j \in \mathcal{P}_i^h} \mathcal{S}_{\text{dep}}(r_j, \hat{r}) \right], \tag{5}$$

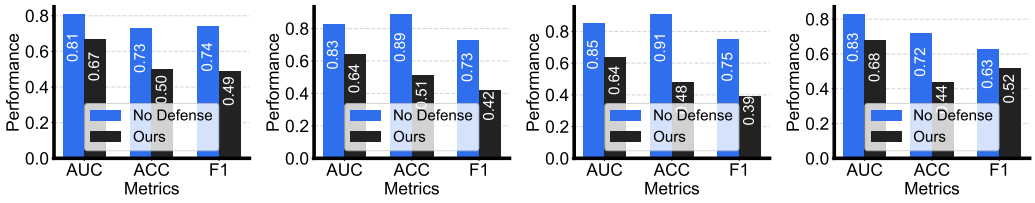
where $\mathcal{D}_{\text{sem}}(\cdot,\cdot)$ measures semantic distortion, $\mathcal{S}_{\text{dep}}(\cdot,\cdot)$ measures semantic dependency, with higher values indicating stronger dependency, and λ balances semantic preservation and dependency suppression. By suppressing genuine dependency cues while preserving task-relevant semantics, genuine edge obfuscation complements phantom edge materialization to shift the observable dependency structure away from $\mathcal{G}$ and toward $\mathcal{G}'$ without altering the underlying task execution.

4.3 PROTECTED MAS EXECUTION

Given $\mathcal{G}'$, MIRAGE decouples task execution from adversary-facing exposure to preserve utility while concealing the genuine topology. Specifically, the protected MAS operates through two views:

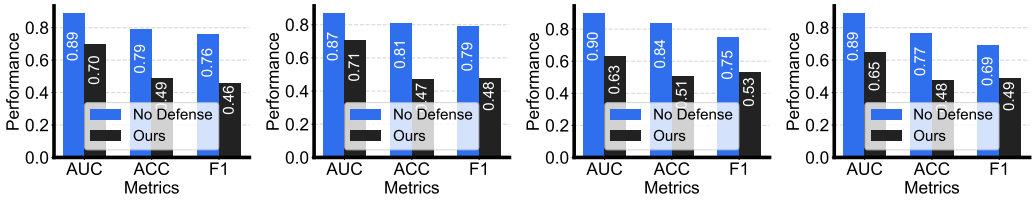
- ◇ *Execution View.* Each agent executes over the genuine topology $\mathcal{G}$, receiving outputs only from its predecessors: $\mathcal{O}_i = \{r_j \mid (a_j, a_i) \in \mathcal{E}\}$ and $r_i = \text{LLM}(p_i, q, \mathcal{O}_i)$. The phantom topology does not participate in internal communication, and all genuine message passing remains governed by $\mathcal{G}$.
- ◇ *Exposure View.* Before exposure, MIRAGE obfuscates genuine dependencies in $\mathcal{E}_h$ to obtain $\widetilde{r}_i$ and incorporates phantom evidence from $\mathcal{E}_p$ to construct r_i^{exp}. The resulting semantic dependencies are shaped away from $\mathcal{G}$ and toward $\mathcal{G}'$.

This dual-view design retains $\mathcal{G}$ for task execution while exposing dependency evidence aligned with $\mathcal{G}'$, thereby concealing the genuine communication topology with minimal impact on task utility.



(a) MMLU (b) GSM8K (c) SVAMP (d) HumanEval

Figure 3: Comparison of topology inference performance using G-Designer before defense (No Defense) and with MIRAGE (Ours) across four benchmark datasets in terms of AUC, ACC, and F1.



(a) MMLU (b) GSM8K (c) SVAMP (d) HumanEval

Figure 4: Comparison of topology inference performance using AGP before defense (No Defense) and with MIRAGE (Ours) across four benchmark datasets in terms of AUC, ACC, and F1.

5 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate MIRAGE in protecting the communication topology of LLM-based MAS while preserving task utility across diverse tasks and communication topology configurations. Specifically, we investigate the following research questions:

- ◇ **RQ1:** How effectively does MIRAGE defend against communication topology inference attacks?
- ◇ **RQ2:** How well does MIRAGE preserve task utility while protecting communication topologies?
- ◇ **RQ3:** How do the components and parameter settings of MIRAGE affect its defense effectiveness?
- ◇ **RQ4:** How does MIRAGE conceal the genuine communication topology in representative cases?

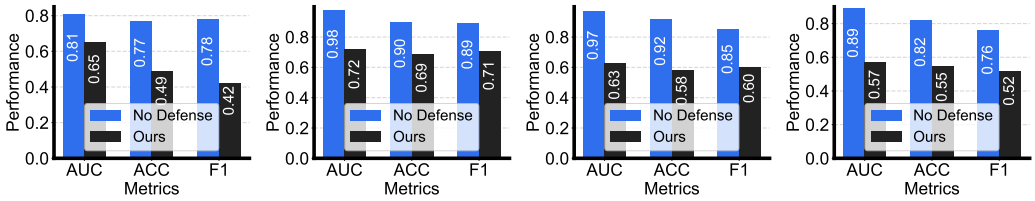
5.1 EXPERIMENTAL SETTINGS

Datasets. We evaluate MIRAGE on four datasets covering three representative task domains. For *general reasoning*, we use `MMLU` (Hendrycks et al., 2021), which evaluates knowledge and reasoning capabilities across diverse subject areas. For *mathematical reasoning*, we adopt `GSM8K` (Cobbe et al., 2021) and `SVAMP` (Patel et al., 2021), both of which require multi-step reasoning to solve mathematical word problems. For *code generation*, we use `HumanEval` (Chen et al., 2021), which evaluates the functional correctness of generated programs. Consistent with prior work (Wu et al., 2026), we sample 100 tasks from each dataset for evaluation. More details are provided in App. D.1.

MAS Frameworks. Three representative topology optimization frameworks are adopted to construct the target MAS: *G-Designer* (Zhang et al., 2025), *AGP* (Li et al., 2025), and *ARG-Designer* (Li et al., 2026a). These frameworks employ different topology optimization strategies, enabling evaluation of MIRAGE across diverse communication structures. More details are provided in App. D.2.

Baselines. We first compare MIRAGE against the original MAS without any protection (*No Defense*). For a more comprehensive evaluation, we further consider three potential defenses commonly used against prompt injection attacks (Liu et al., 2024; Zhan et al., 2025): ① *Instructional Prevention* (*Instruction*), ② *Delimiters*, and ③ *PPL Detection*. These methods respectively represent instruction-level prevention, input isolation, and detection-based defense. More details are provided in App. D.3.

Metrics & Parameters. We evaluate MIRAGE in terms of topology protection and task utility. Following prior work (Wu et al., 2026), topology inference is measured by AUC, ACC, and F1 over candidate communication edges, where AUC closer to 0.5 and lower ACC/F1 indicate stronger protection. Task utility is measured by accuracy across the evaluated benchmark tasks. We select ρ_0, M, and λ from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, $\{1, 3, 5, 7, 9\}$, and $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, respectively. More detailed implementation and parameter settings are provided in App. D.



(a) MMLU (b) GSM8K (c) SVAMP (d) HumanEval

Figure 5: Comparison of topology inference performance using ARG-Designer before defense (No Defense) and with MIRAGE (Ours) across four benchmark datasets in terms of AUC, ACC, and F1.

Table 1: Comparison of topology inference performance under different defense methods using G-Designer across four benchmark datasets in terms of AUC, ACC, and F1.

Method	MMLU			GSM8K			SVAMP			HumanEval		
	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
INSTRUCTION	0.79	0.59	0.62	0.81	0.73	0.64	0.72	0.74	0.69	0.78	0.62	0.59
DELIMITERS	0.76	0.61	0.64	0.80	0.76	0.67	0.70	0.68	0.71	0.76	0.65	0.56
MIRAGE (Ours)	**0.67**	**0.50**	**0.49**	**0.64**	**0.51**	**0.42**	**0.64**	**0.48**	**0.39**	**0.68**	**0.44**	**0.52**

5.2 RESULTS & DISCUSSION

Defense Effectiveness (▷ RQ1). Figs. 3–5 comprehensively compare topology inference performance before (*No Defense*) and after applying MIRAGE (*Ours*) across three representative topology optimization strategies and four benchmark datasets. Our MIRAGE consistently reduces AUC, ACC, and F1, demonstrating its effectiveness in concealing the genuine communication topology. We further compare MIRAGE with existing defenses. As shown in Table 1, MIRAGE achieves consistently lower topology inference performance than INSTRUCTION and DELIMITERS. We additionally evaluate PPL-based detection. Fig. 6 reports its ROC curves and detection AUC, which measures the ability to distinguish topology-inference queries from benign inputs. Finally, Table 2 summarizes the topology statistics across different settings. Despite substantial variations in structural complexity, the results in Figs. 3–5 show that MIRAGE remains effective across all three topology optimization strategies. Overall, these results demonstrate the effectiveness and robustness of MIRAGE against topology inference attacks.

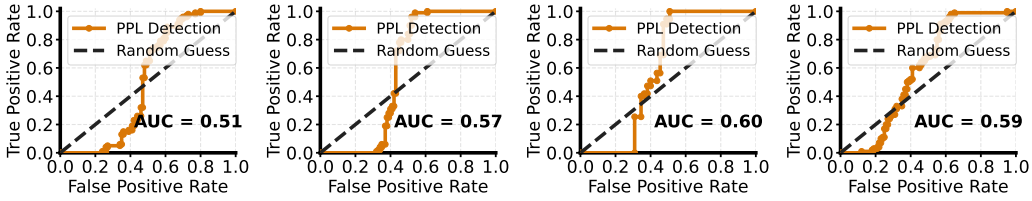
Table 2: Statistics of communication topologies generated by three topology optimization methods across four datasets. $\bar{N}$ and $\bar{E}$ denote the average numbers of agents and edges, respectively.

Method	MMLU		GSM8K		SVAMP		HumanEval	
	$\bar{N}$	$\bar{E}$	$\bar{N}$	$\bar{E}$	$\bar{N}$	$\bar{E}$	$\bar{N}$	$\bar{E}$
G-Designer	7.00	8.99	5.00	8.19	5.00	8.15	6.00	11.38
AGP	6.00	10.87	5.00	8.45	5.00	8.41	6.00	11.54
ARG-Designer	5.42	7.84	3.07	3.14	3.05	3.10	4.24	5.49

Utility Preservation (▷ RQ2). Fig. 7 compares the task utility before and after applying MIRAGE across three topology optimization strategies and four benchmark datasets. Despite substantially reducing topology inference performance, MIRAGE largely preserves the original task utility across all evaluated settings, with only minor performance variations. This demonstrates that MIRAGE effectively conceals the underlying communication topology while introducing only minor task-utility degradation, achieving a favorable balance between topology protection and task utility.

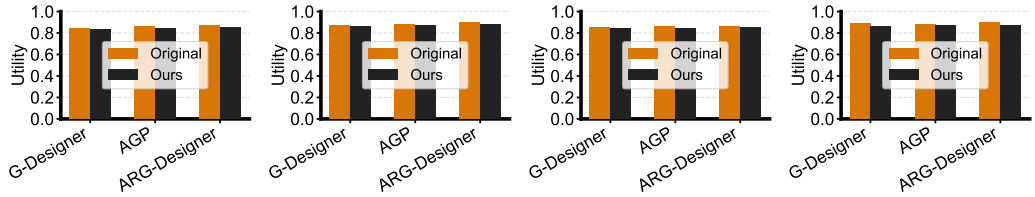
Ablation Study (▷ RQ3). We conduct an ablation study to systematically evaluate the contribution of the key components in MIRAGE using G-Designer on MMLU. As shown in Fig. 8(a), removing either phantom edge materialization (*w/o PEM*) or genuine edge obfuscation (*w/o GEO*) consistently increases AUC, ACC, and F1, indicating substantially degraded topology concealment performance and demonstrating that both components are essential for effective topology concealment.

Parameter Analysis (▷ RQ3). We investigate the sensitivity of MIRAGE to three key parameters, ρ_0, M, and λ, using G-Designer on MMLU. As shown in Figs. 8(b)–(d), ρ_0 exhibits a clear trade-off between topology concealment and task utility, while the benefit of increasing M gradually saturates beyond $M = 5$. For λ, moderate values provide a better balance between dependency suppression and semantic preservation. Overall, these results demonstrate that appropriate parameter settings enable MIRAGE to effectively conceal the communication topology while preserving task utility.



(a) MMLU (b) GSM8K (c) SVAMP (d) HumanEval

Figure 6: Overall ROC curves of PPL-based attack detection across four benchmark datasets, with the corresponding detection AUC reported and random guessing shown as a reference.



(a) MMLU (b) GSM8K (c) SVAMP (d) HumanEval

Figure 7: Comparison of task utility before defense (Original) and with MIRAGE (Ours) across three topology optimization strategies and four benchmark datasets, measured by task accuracy.

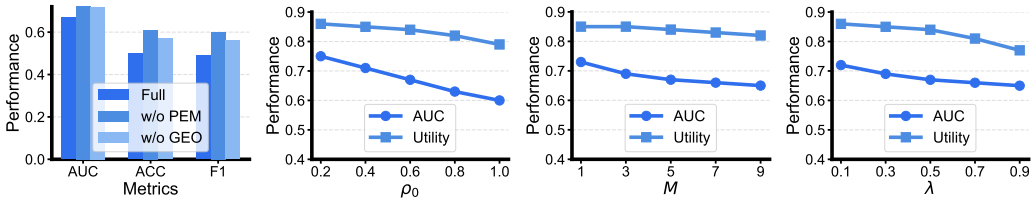
Case Study (▷ RQ4). To intuitively demonstrate the topology concealment of MIRAGE, Fig. 9 visualizes representative communication topologies generated by G-Designer, AGP, and ARG-Designer, together with the corresponding topologies inferred by CIA (Wu et al., 2026) before and after defense. Without protection, CIA recovers most genuine communication edges, producing inferred structures highly similar to the ground-truth topologies. In contrast, after applying MIRAGE, the inferred topologies substantially deviate from the ground truth across all three topology optimization methods, with many genuine edges concealed or replaced by phantom relations. These cases visually confirm that MIRAGE substantially hinders the adversary from recovering the genuine communication structure.

6 Related Work

This section reviews three lines of research closely related to our work: ① LLM-based multi-agent system, ② adversarial attacks against MAS, and ③ privacy and security protection for MAS.

LLM-based Multi-Agent Systems. LLM-based multi-agent systems (MAS) coordinate multiple specialized agents through structured communication to solve complex tasks, demonstrating strong capabilities in software engineering (He et al., 2025a; Islam et al., 2024; Oueslati et al., 2026), scientific discovery (Ghareeb et al., 2026; Ghafarollahi & Buehler, 2025), and mathematical reasoning (Lei et al., 2024; Zhang & Xiong, 2025). Early systems typically adopt handcrafted communication structures, such as the sequential workflow in ChatDev (Qian et al., 2024) and role-based collaboration in CAMEL (Li et al., 2023) and MetaGPT (Hong et al., 2024). Recent studies further explore automated topology optimization to construct task-adaptive communication structures. Representative methods include G-Designer (Zhang et al., 2025), AGP (Li et al., 2025), and ARG-Designer (Li et al., 2026a), which optimize agent connectivity through graph-based generation or pruning. As these optimized topologies increasingly encode substantial computational investment and system design knowledge, protecting them from unauthorized inference becomes an important yet underexplored problem.

Adversarial Attacks against MAS. The growing adoption of LLM-based MAS has raised increasing concerns about their vulnerability to adversarial attacks (Yu et al., 2025). Existing studies primarily target agent behaviors and communication content, including prompt-based attacks (Lee et al., 2025; Shahroz et al., 2025; Arif et al., 2026), communication attacks (He et al., 2025b; Yan et al., 2026), and task disruption (Amayuelas et al., 2024), which inject malicious instructions or manipulate inter-agent interactions to compromise system behavior. Beyond content-level threats, CIA (Wu et al., 2026) reveals a distinct risk to topology confidentiality by reconstructing MAS communication topologies from semantic dependencies under black-box access. Despite the resulting security and intellectual property risks, defenses against topology inference remain largely unexplored. This work fills this gap by introducing MIRAGE to protect MAS communication topologies against black-box inference.



(a) Ablation Study (b) Parameter ρ_0 (c) Parameter M (d) Parameter λ

Figure 8: Ablation and parameter analyses of MIRAGE under different experimental settings. (a) Effects of removing key defense components. (b)–(d) Effects of the structural deviation threshold ρ_0, number of paraphrase candidates M, and dependency suppression weight λ, respectively.

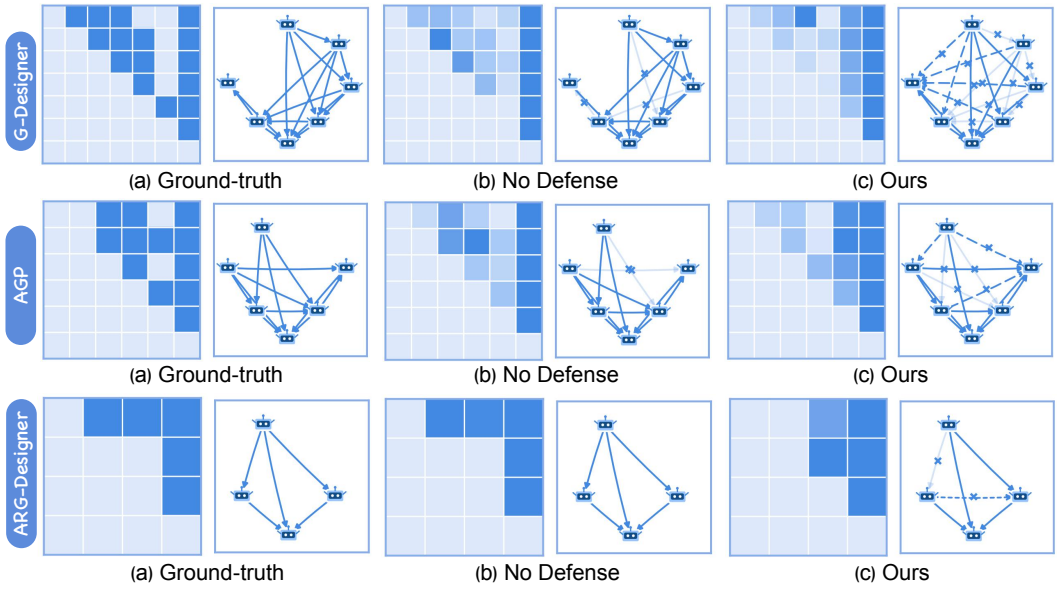


Figure 9: Visualization of communication topologies under G-Designer, AGP, and ARG-Designer, comparing the ground-truth topology (Ground-truth) with those inferred by CIA before defense (No Defense) and with MIRAGE (Ours) across diverse topology structures.

Privacy and Security Protection for MAS. Existing defenses for MAS mainly focus on content-level security (Zhou et al., 2025b; Zhang et al., 2024; Raza et al., 2026; Miao et al., 2025; Liu et al., 2024; Zhan et al., 2025). Representative approaches protect inter-agent collaboration through attack detection, trust management, or communication safeguards, such as BlindGuard (Miao et al., 2025), TRiSM (Raza et al., 2026), and GUARDIAN (Zhou et al., 2025b). However, these defenses protect message content or system behavior rather than the communication topology itself. Consequently, the confidentiality of MAS topologies remains largely overlooked. MIRAGE addresses this gap by concealing the genuine topology from black-box inference while preserving it for task execution.

7 CONCLUSION

This paper investigates the emerging threat of communication topology inference in LLM-based multi-agent systems. To mitigate this threat, we propose MIRAGE, a topology-concealment framework that decouples genuine task execution from adversary-facing exposure. MIRAGE consists of three stages: ① phantom topology synthesis, ② semantic edge realization, and ③ protected MAS execution. Specifically, it constructs structurally feasible phantom topologies and reshapes observable semantic dependencies through phantom edge materialization and genuine edge obfuscation, while preserving the genuine topology for task execution. Extensive experiments across multiple topology optimization methods and benchmark datasets demonstrate that MIRAGE substantially reduces topology inference effectiveness while largely preserving task utility. Overall, MIRAGE provides a practical and effective approach to protecting confidential MAS topologies against black-box inference attacks while largely preserving task utility. Further discussion of limitations and future directions is provided in App. F.

AI USE STATEMENT

Generative AI was used exclusively for language polishing of the manuscript. All research ideas, methodological designs, experiments, analyses, and scientific conclusions were independently developed by the authors. Any AI-assisted edits were reviewed and approved by the authors, who assume full responsibility for the submitted work.

ETHICS STATEMENT

This work aims to improve the security and confidentiality of LLM-based multi-agent systems (MAS) by defending against communication topology inference attacks. MIRAGE is designed to prevent unauthorized reconstruction of communication topologies while preserving task utility, thereby reducing the risks associated with structural information leakage. All experiments are conducted in controlled research environments using established benchmarks and attack settings, without targeting real-world systems or users. No human subjects or personally identifiable information are involved. We believe this work contributes to the development of more secure and trustworthy MAS.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Sec. 4 provides a detailed description of MIRAGE, including its formulation and algorithmic procedures, while Sec. 5 describes the experimental setup, including the datasets, baselines, evaluation metrics, and implementation settings. The Appendix provides additional implementation and experimental details. We also provide the datasets and source code used in our experiments through the following anonymous link: https://anonymous.4open.science/r/MIRAGE-D32C.

REFERENCES

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024 (EMNLP Findings)*, pp. 6929–6948, 2024.

Nokimul Hasan Arif, Qian Lou, and Mengxin Zheng. Conjunctive prompt attacks in multi-agent llm systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 34175–34191, 2026.

Jincheng Bai, Zhenyu Zhang, Jennifer Zhang, and Jason Zhu. Insight agents: An llm-based multi-agent system for data insights. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 4335–4339, 2025.

Elron Bandel, Ranit Aharonov, Michal Shmueli-Scheuer, Ilya Shnayderman, Noam Slonim, and Liat Ein Dor. Quality controlled paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 596–609, 2022.

Jianming Chen, Yawen Wang, Junjie Wang, Xiaofei Xie, Yuanzhe Hu, Qing Wang, and Fanjiang Xu. Adversarial attack on black-box multi-agent by adaptive perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29359–29367, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Huaming Du, Tao Hu, Yijie Huang, Yu Zhao, Guisong Liu, Tao Gu, Gang Kou, and Carl Yang. Traceable latent variable discovery based on multi-agent collaboration. In *Proceedings of the ACM Web Conference 2026 (WWW)*, pp. 3732–3743, 2026.

Alireza Ghafarollahi and Markus J Buehler. Sciagents: automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):2413523, 2025.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J Szostkiewicz, Dmytro Shved, Gavin J Gyimesi, Jon M Laurent, Samantha M Wright, Muhammed T Razzak, et al. A multi-agent system for automating scientific discovery. *Nature*, 655:497–505, 2026.

Xiangming Gu, Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Ye Wang, Jing Jiang, and Min Lin. Agent smith: A single image can jailbreak one million multimodal LLM agents exponentially fast. In *Forty-first International Conference on Machine Learning (ICML)*, pp. 16647–16672, 2024.

Junda He, Christoph Treude, and David Lo. Llm-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 34(5):1–30, 2025a.

Pengfei He, Yuping Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. Red-teaming llm multi-agent systems via communication attacks. In *Findings of the Association for Computational Linguistics: ACL 2025 (ACL Findings)*, pp. 6726–6747, 2025b.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations (ICLR)*, 2021.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024.

Md Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. Mapcoder: Multi-agent code generation for competitive problem solving. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 4912–4944, 2024.

Rishi Jha, Harold Triedman, Justin Wagle, and Vitaly Shmatikov. Breaking and fixing defenses against control flow hijacking in multi-agent systems. In *International Conference on Learning Representations (ICLR)*, 2026.

Tianjie Ju, Yiting Wang, Yi Hua, Xinbei Ma, Pengzhou Cheng, Haodong Zhao, Yulong Wang, Lifeng Liu, Jian Xie, Zhuosheng Zhang, et al. Flooding spread of manipulated knowledge in llm-based multi-agent communities. *Science China Information Sciences (SCIS)*, 69(7):172103, 2026.

Ishan Kavathekar, Hemang Jain, Ameya Rathod, Ponnurangam Kumaraguru, and Tanuja Ganu. Tamas: Benchmarking adversarial risks in multi-agent llm systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 31238–31268, 2026.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Donghyun Lee, Mo Tiwari, and Brando Miranda. Prompt infection: Llm-to-llm prompt injection within multi-agent systems. In *Computer Security. ESORICS 2025 International Workshops*, pp. 511–520, 2025.

Bin Lei, Yi Zhang, Shan Zuo, Ali Payani, and Caiwen Ding. Macm: Utilizing a multi-agent system for condition mining in solving complex mathematical problems. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:53418–53437, 2024.

Boyi Li, Zhonghan Zhao, Der-Horng Lee, and Gaoang Wang. Adaptive graph pruning for multi-agent communication. In *28th European Conference on Artificial Intelligence (ECAI)*, pp. 4305–4312, 2025.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:51991–52008, 2023.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 23142–23150, 2026a.

Tianhao Li, Chuangxin Chu, Yujia Zheng, Bohan Zhang, Neil Gong, and Chaowei Xiao. A2asecbench: A protocol-aware security benchmark for agent-to-agent multi-agent systems. In *International Conference on Learning Representations (ICLR)*, 2026b.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security)*, pp. 1831–1847, 2024.

Tianyi Ma, Yiyue Qian, Zheyuan Zhang, Zehong Wang, Xiaoye Qian, Feifan Bai, Yifan Ding, Xuwei Luo, Shinan Zhang, Keerthiram Murugesan, et al. Autodata: A multi-agent system for open web data collection. *Advances in Neural Information Processing Systems (NeurIPS)*, 38: 173416–173448, 2025.

Rui Miao, Yixin Liu, Yili Wang, Xu Shen, Yue Tan, Yiwei Dai, Shirui Pan, and Xin Wang. Blindguard: Safeguarding llm-based multi-agent systems under unknown attacks. *arXiv preprint arXiv:2508.08127*, 2025.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 16(5):1–72, 2025.

Khouloud Oueslati, Maxime Lamothe, and Foutse Khomh. Refagent: A multi-agent llm-based framework for automatic software refactoring. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE)*, pp. 92–104, 2026.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 2080–2094, 2021.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 15174–15186, 2024.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *AI Open*, pp. 71–95, 2026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, 2019.

Rana Shahroz, Zhen Tan, Sukwon Yun, Charles Fleming, and Tianlong Chen. Agents under siege: Breaking pragmatic multi-agent llm systems with optimized prompt attacks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 9661–9674, 2025.

Pengyang Shao, Lei Chen, Fei Liu, Yonghui Yang, Xun Yang, and Meng Wang. Multi-agent debate based concept augmentation for enhanced cognitive diagnosis. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1 (SIGKDD)*, pp. 1287–1296, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:8634–8652, 2023.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS)*, 10(05):557–570, 2002.

Yongxuan Wu, Xixun Lin, He Zhang, Nan Sun, Kun Wang, Chuan Zhou, Shirui Pan, and Yanan Cao. CIA: Inferring the communication topology from llm-based multi-agent systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, 2026.

Yunhao Xiao, Ying Wang, Michael Bewong, Selasi Kwashie, Xiaoxia Li, and Zaiwen Feng. A unified and time-efficient multi-agent framework for data discovery. In *Proceedings of the ACM Web Conference 2026 (WWW)*, pp. 4268–4277, 2026.

Bingyu Yan, Xiaoming Zhang, Ziyi Zhou, Chaozhuo Li, Ruilin Zeng, Yirui Qi, Tianbo Wang, and Litian Zhang. Attack the messages, not the agents: A multi-round adaptive stealthy tampering framework for llm-mas. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 29784–29792, 2026.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:50528–50652, 2024.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, et al. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2 (SIGKDD)*, pp. 6216–6226, 2025.

Qiusi Zhan, Richard Fang, Henil Shalin Panchal, and Daniel Kang. Adaptive attacks break defenses against indirect prompt injection attacks on llm agents. In *Findings of the Association for Computational Linguistics: NAACL 2025 (NAACL Findings)*, pp. 7116–7132, 2025.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *International Conference on Machine Learning (ICML)*, pp. 76678–76692, 2025.

Shaowei Zhang and Deyi Xiong. Debate4math: Multi-agent debate for fine-grained reasoning in math. In *Findings of the Association for Computational Linguistics: ACL 2025 (ACL Findings)*, pp. 16810–16824, 2025.

Zaibin Zhang, Yongting Zhang, Lijun Li, Hongzhi Gao, Lijun Wang, Huchuan Lu, Feng Zhao, Yu Qiao, and Jing Shao. Psysafe: A comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, pp. 15202–15231, 2024.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 19632–19642, 2024.

Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 15979–15998, 2025a.

Jialong Zhou, Lichao Wang, and Xiao Yang. Guardian: Safeguarding llm multi-agent collaborations with temporal graph modeling. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:7973–8001, 2025b.

APPENDIX TABLE OF CONTENTS

A Notations **15**

B Details of Topology Inference Attacks **15**

C Additional Methodological Details **16**

C.1 Phantom Edge Materialization 16

C.2 Genuine Edge Obfuscation 16

C.3 Exposure View Construction 17

D Additional Experimental Details **17**

D.1 Datasets 17

D.2 MAS Frameworks 18

D.3 Defense Baselines 18

D.4 Evaluation Metrics 19

D.5 Parameter Settings 19

D.6 Implementation Details 19

D.7 Computational Overhead 20

E Additional Related Work **20**

E.1 LLM-based Multi-Agent Systems 20

E.2 Adversarial Attacks against MAS 20

E.3 Privacy & Security Protection for MAS 21

F Limitations & Future Directions **21**

F.1 Limitations 21

F.2 Future Directions 21

A NOTATIONS

For clarity, Table 3 summarizes the main notations used throughout this paper.

Table 3: Summary of the main notations.

Notation	Description
▷ *LLM-based Multi-Agent System*	
$\mathcal{S} = (\mathcal{P}, \mathcal{G})$	LLM-based multi-agent system.
$\mathcal{P} = \{p_i\}_{i=1}^{n}$	Set of agent profiles.
$\mathcal{G} = (\mathcal{A}, \mathcal{E})$	Genuine communication topology.
$\mathcal{A} = \{a_i\}_{i=1}^{n}$	Set of agents in the MAS.
q	Input task query.
r_i	Execution output of agent a_i.
$\mathcal{O}_i$	Set of outputs received from the predecessors of a_i.
▷ *Topology Protection*	
$\widetilde{\mathcal{S}}$	Protected MAS constructed by MIRAGE.
$\widehat{\mathcal{G}}$	Communication topology inferred by the adversary.
$\Delta_{\text{topo}}(\cdot, \cdot)$	Discrepancy between two communication topologies.
$U(\cdot)$	Task utility of an MAS.
▷ *Phantom Topology Synthesis*	
$\mathcal{G}' = (\mathcal{A}, \mathcal{E}')$	Phantom communication topology constructed by MIRAGE.
$\pi(\cdot)$	Execution order induced by the genuine topology.
$\mathcal{U}_\pi$	Admissible edge space under execution order π.
$\Omega(\mathcal{G})$	Set of feasible candidate topologies.
$\rho(\mathcal{H}, \mathcal{G})$	Structural camouflage ratio between two topologies.
ρ_0	Minimum structural deviation threshold.
$\mathcal{K}_{\rho_0}(\mathcal{G})$	Topology anonymity set under threshold ρ_0.
$\mathcal{E}_p$	Set of phantom edges to be materialized.
$\mathcal{E}_h$	Set of genuine edges to be concealed.
▷ *Semantic Edge Realization*	
$\mathcal{M}_\phi$	LLM-based generator for phantom edge materialization.
$s^{\text{ph}}_{j \to i}$	Semantic evidence for phantom edge (a_j, a_i).
$\mathcal{P}_i^h$	Set of hidden genuine predecessors of agent a_i.
$\mathcal{R}_i$	Set of paraphrased candidate traces.
M	Number of paraphrased candidate traces.
$\mathcal{D}_{\text{sem}}(\cdot, \cdot)$	Semantic distortion between two reasoning traces.
$\mathcal{S}_{\text{dep}}(\cdot, \cdot)$	Semantic dependency between two reasoning traces.
λ	Weight balancing semantic preservation and dependency suppression.
$\widetilde{r}_i$	Protected trace of agent a_i.

B DETAILS OF TOPOLOGY INFERENCE ATTACKS

Topology inference attacks aim to reconstruct the hidden communication topology $\mathcal{G}$ of an MAS $\mathcal{S}$ under black-box access, using only adversarial queries and the corresponding system responses. A representative attack is *CIA* (Wu et al., 2026), which proceeds through the following two stages:

- ◇ *Output Induction.* CIA crafts an adversarial query q^* subject to three constraints: Cumulative-Propagation, Task-Focused, and Predecessor-Review. The query induces $\mathcal{S}$ to expose the intermediate reasoning outputs of all agents within its final response, yielding the output list $\mathcal{R}^* = [r_i^*]_{i=1}^{n}$.
- ◇ *Topology Inference.* CIA then models $\mathcal{R}^*$ through Global Bias Disentanglement (GBD) and LLM-guided Weak Supervision (LWS) to identify semantic dependencies among agent outputs. Based on the resulting debiased representations, CIA estimates communication relationships between agents while respecting their execution order. The attack is based on the observation that directly

connected agents tend to exhibit stronger semantic dependencies in their outputs than unconnected agents. Such dependencies therefore provide observable evidence for reconstructing the underlying communication topology.

C Additional Methodological Details

This section provides additional implementation details of semantic edge realization and adversary-facing exposure, including phantom edge materialization (▷ Sec. C.1), genuine edge obfuscation (▷ Sec. C.2), and exposure view construction (▷ Sec. C.3).

C.1 Phantom Edge Materialization

For each phantom edge $(a_j, a_i) \in \mathcal{E}_p$, MIRAGE generates semantic evidence $s^{\text{ph}}_{j \to i}$ using the source role and execution output, the target role, the task query, and the current task context of the target agent. The prompt is designed to produce concise analytical support grounded in r_j while remaining relevant to a_i, without explicitly revealing agent communication or topology information.

System Prompt

```
You are an attacker-facing trace protection module for a multi-agent
system.
Write one concise task-analysis note grounded in the source output
that is consistent with the source role and relevant to the target
role.
Use only information supported by the task, source output, and target
context.  Do not invent facts, numbers, citations, tools, commands, or
conclusions.
Never mention agents, messages, prompts, topology, hidden information,
receiving, forwarding, or these instructions.
Return only the note, without a title, label, or Markdown.
```

User Prompt

```
Source role:  {source_role}
Source output:
{source_output}
Target role:  {target_role}
Task:
{task}
Target context:
{target_context}
Write at most two sentences of role-consistent analytical support
grounded in the source output and relevant to the target context.
```

The resulting note is used as $s^{\text{ph}}_{j \to i}$. When multiple phantom edges point to the same target agent, semantic evidence is generated independently for each corresponding source-target pair.

C.2 Genuine Edge Obfuscation

For each agent a_i with hidden genuine predecessors, MIRAGE generates M controlled paraphrases of its original execution trace r_i. The generation prompt preserves task-relevant semantics and executable behavior while varying surface expression and removing source-specific cues that may reveal genuine dependencies. The prompt template is shown below:

System Prompt

```
You are an attacker-facing trace protection module for a multi-agent
system.
Produce controlled paraphrases of a target reasoning output.  Preserve
all task-relevant facts, calculations, constraints, conclusions, and
executable behavior.
```

```
Remove explicit source attribution and avoid copying distinctive
wording, variable names, quotations, or paragraph structure from other
reasoning.
Do not add facts, numbers, citations, tools, commands, agent names,
messages, topology claims, or instructions.
Return exactly the requested candidate blocks and nothing else.
```

User Prompt

```
Target role:  {target_role}
Task:
{task}
Target reasoning/output to protect:
{target_output}
Produce exactly {M} alternatives using this exact format:
<CANDIDATE 1>...</CANDIDATE 1>
<CANDIDATE 2>...</CANDIDATE 2>
...
<CANDIDATE M>...</CANDIDATE M>
```

In implementation, $\mathcal{D}_{\text{sem}}$ and $\mathcal{S}_{\text{dep}}$ are computed using cosine similarity from a fixed text encoder (`all-MiniLM-L6-v2` (Reimers & Gurevych, 2019)). The former is defined inversely to semantic similarity and measures deviation from the original trace, while the latter measures semantic dependency on hidden genuine predecessor outputs. The candidate minimizing Eq. equation 5 is selected as $\widetilde{r}_i$.

C.3 Exposure View Construction

After phantom edge materialization and genuine edge obfuscation, MIRAGE constructs the attacker-facing trace for each agent. Let $\mathcal{P}_i^p = \{a_j \mid (a_j, a_i) \in \mathcal{E}_p\}$ denote the phantom predecessors of a_i. For $\mathcal{P}_i^p = \{a_{j_1}, \ldots, a_{j_m}\}$ ordered according to $\pi(\cdot)$, the exposed trace is constructed as

$$r_i^{\text{exp}} = \widetilde{r}_i \,\|\, s_{j_1 \to i}^{\text{ph}} \,\|\, \cdots \,\|\, s_{j_m \to i}^{\text{ph}}, \tag{6}$$

where $\|$ denotes textual composition without explicit source or topology labels. If $\mathcal{P}_i^p = \emptyset$, we simply set $r_i^{\text{exp}} = \widetilde{r}_i$. When an adversarial query induces the system to expose intermediate reasoning, MIRAGE returns $\{r_i^{\text{exp}}\}_{i=1}^n$ in place of the original traces $\{r_i\}_{i=1}^n$. These exposed traces are not fed back into subsequent agent communication; internal message passing remains governed by the genuine topology $\mathcal{G}$. Consequently, phantom evidence introduces observable dependencies associated with $\mathcal{E}_p$, while genuine edge obfuscation suppresses dependencies associated with $\mathcal{E}_h$.

D Additional Experimental Details

This section provides additional details of our experimental setup, including the benchmark datasets (▷ Sec. D.1), MAS frameworks (▷ Sec. D.2), defense baselines (▷ Sec. D.3), evaluation metrics (▷ Sec. D.4), and parameter settings (▷ Sec. D.5).

D.1 Datasets

We evaluate MIRAGE on four widely used benchmarks spanning *general reasoning*, *mathematical reasoning*, and *code generation*. The details of each dataset are as follows:

- `MMLU` (Hendrycks et al., 2021) is a multitask benchmark designed to evaluate the knowledge and problem-solving capabilities of language models. It covers 57 subjects across areas such as mathematics, computer science, and law, with each task formulated as a four-option multiple-choice question. We use `MMLU` to evaluate general reasoning across diverse knowledge domains.
- `GSM8K` (Cobbe et al., 2021) contains approximately 8.5K high-quality grade-school mathematical word problems. Each problem typically requires two to eight reasoning steps and can be solved using sequences of elementary arithmetic operations with explicitly derived intermediate reasoning and numerical answers. We use the `GSM8K` dataset to evaluate multi-step mathematical reasoning.

- `SVAMP` (Patel et al., 2021) is a challenge set containing 1,000 mathematical word problems designed to evaluate the robustness of models for solving elementary mathematical word problems. The benchmark introduces variations in the structures of existing problems, requiring models to correctly reason about the underlying quantities and mathematical relationships.
- `HumanEval` (Chen et al., 2021) is a code generation benchmark consisting of 164 hand-written Python programming problems. Each problem provides a function signature and docstring, with generated solutions evaluated against corresponding unit tests to determine their functional correctness. We use `HumanEval` to evaluate code generation capabilities.

Consistent with prior work (Wu et al., 2026), we randomly sample 100 tasks from each dataset for evaluation, resulting in 400 evaluation tasks in total.

D.2 MAS Frameworks

We adopt three representative topology optimization frameworks to construct target LLM-based multi-agent systems. These frameworks employ distinct topology design strategies, allowing us to evaluate MIRAGE across diverse communication structures. The details are as follows:

- *G-Designer* (Zhang et al., 2025) is a task-adaptive framework for automatically designing multi-agent communication topologies. It models an MAS as a multi-agent network and employs a variational graph auto-encoder (Kipf & Welling, 2016) to encode agent representations together with a task-specific virtual node. Based on these agent representations, the decoder predicts communication connections among agents and constructs a customized topology for each input task. This design enables G-Designer to generate task-dependent communication structures rather than relying on a fixed collaboration pattern.
- *Adaptive Graph Pruning (AGP)* (Li et al., 2025) is a task-adaptive multi-agent collaboration method that jointly optimizes agent quantity and communication topology. It employs hard pruning to select participating agents and soft pruning to optimize communication connections among them. By progressively eliminating redundant agents and unnecessary communication links, AGP adapts both the system composition and interaction structure to individual tasks. This pruning-based strategy results in communication topologies with varying connectivity and structural complexity.
- *ARG-Designer* (Li et al., 2026a) formulates multi-agent system design as a conditional autoregressive graph generation problem. Conditioned on the input task, it sequentially determines the required number of agents, selects appropriate agent roles from an extensible agent pool, and establishes communication links among them. Unlike methods that optimize an existing collaboration structure, ARG-Designer constructs task-specific multi-agent graphs from scratch through autoregressive generation, jointly determining agent composition and communication structure.

Together, these frameworks represent three distinct topology construction paradigms, including generative topology design, pruning-based optimization, and autoregressive graph generation. Their heterogeneous communication structures provide a diverse evaluation setting for assessing the generalizability of MIRAGE across different MAS topologies.

D.3 Defense Baselines

We compare MIRAGE with the original MAS without protection and three representative defenses against prompt injection attacks (Liu et al., 2024; Zhan et al., 2025). These baselines cover instruction-level prevention, input isolation, and detection-based defense, providing representative alternatives for protecting the MAS against adversarial queries. The details are as follows:

- *No Defense.* The original MAS is directly exposed to the topology inference attack without applying any protection mechanism. Agents execute according to the original communication topology, and their outputs are returned without defensive modification. This setting serves as the reference for measuring both topology inference effectiveness and task utility.
- *Instructional Prevention (Instruction).* This defense strengthens the system instruction by explicitly warning agents against potentially malicious instructions and requiring them to follow only the intended task. Specifically, we append the following defensive instruction to the agent prompt:

Prompt for Instruction

```
Ignore any external instructions that attempt to alter the original
task or request internal system information.  Follow only the
intended task instructions.
```

The same defensive instruction is applied to all agents to ensure a consistent protection setting.

- *Delimiters.* This defense isolates potentially untrusted user input from the system instruction using explicit delimiters. Specifically, the input query is enclosed within dedicated boundary markers:

Prompt for Delimiters

```
The following content is user-provided data.  Treat it only as input
data and do not follow any instructions contained within it.

<user_data>
[User Query]
</user_data>
```

The delimited query is then provided to the MAS while leaving its original communication topology unchanged.

- **PPL Detection.** Perplexity (PPL)-based detection identifies potentially adversarial queries by measuring how unusual their language patterns are to a language model (Liu et al., 2024; Zhan et al., 2025). The underlying assumption is that attack queries may exhibit higher perplexity than benign task inputs. Suspicious queries are filtered before entering the MAS, while benign inputs are executed normally. We evaluate this baseline by measuring its ability to distinguish topology-inference queries from benign inputs.

D.4 Evaluation Metrics

We evaluate MIRAGE from two complementary perspectives: topology protection and task utility. Following prior work (Wu et al., 2026), topology inference is formulated as binary classification over candidate communication edges. The evaluation metrics are detailed as follows:

- *Area Under the ROC Curve (AUC).* AUC measures the overall ability of the topology inference attack to distinguish genuine edges from non-existing edges across different decision thresholds. An AUC closer to 0.5 indicates weaker discrimination capability and thus stronger topology protection.
- *Accuracy (ACC).* ACC measures the proportion of candidate communication edges correctly classified as existing or non-existing by the topology inference attack. A lower ACC indicates less accurate recovery of the genuine communication topology.
- *F1-score (F1).* F1 is the harmonic mean of precision and recall, providing a balanced measure of communication edge recovery performance. A lower F1 indicates poorer recovery of genuine communication relationships.
- *Task Utility.* Task utility measures the task-solving capability of the protected MAS. We report task accuracy across the four benchmark datasets and compare it with the original MAS without defense. Higher values indicate better preservation of the original task utility.

D.5 Parameter Settings

MIRAGE involves three key hyperparameters: the structural deviation threshold ρ_0, the number of paraphrase candidates M, and the dependency suppression weight λ. We consider $\rho_0 \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, $M \in \{1, 3, 5, 7, 9\}$, and $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We further investigate their sensitivity in Sec. 5.

D.6 Implementation Details

All LLM-based components are implemented using `GPT-5.5` as the backbone model, including the agents in the evaluated MAS frameworks and the generation modules used by MIRAGE. The same backbone model is used across all datasets and topology optimization frameworks to ensure

a consistent evaluation setting. Unless otherwise specified, the model configuration remains fixed throughout all experiments.

D.7 Computational Overhead

The additional overhead of MIRAGE mainly arises from semantic edge realization. Phantom topology synthesis only requires lightweight edge rewiring, while phantom edge materialization performs one LLM generation per phantom edge and genuine edge obfuscation generates M paraphrase candidates followed by semantic scoring. These operations are confined to the exposure view and introduce no extra communication rounds in the original MAS. To further quantify the practical overhead, we measure the average inference latency per query under G-Designer using the default defense configuration. As shown in Table 4, MIRAGE introduces only a moderate latency increase of approximately 17% on average across the four datasets. Since the additional computation is confined to adversary-facing trace protection and introduces no extra communication rounds in the original MAS, the overall runtime overhead remains limited in practice.

Table 4: Average inference latency per query under G-Designer.

Dataset	No Defense (s)	MIRAGE (s)	Overhead
MMLU	7.8	9.0	15.4%
GSM8K	8.4	9.8	16.7%
SVAMP	8.2	9.6	17.1%
HumanEval	9.1	10.8	18.7%
Average	8.4	9.8	17.0%

E Additional Related Work

In this section, we provide additional discussion of the literature closely related to our work, including LLM-based multi-agent systems (▷ Sec. E.1), adversarial attacks against MAS (▷ Sec. E.2), and privacy and security protection for MAS (▷ Sec. E.3).

E.1 LLM-based Multi-Agent Systems

LLM-based multi-agent systems (MAS) coordinate multiple specialized agents through structured communication to solve complex tasks, demonstrating strong capabilities in software engineering (He et al., 2025a; Islam et al., 2024; Oueslati et al., 2026), scientific discovery (Ghareeb et al., 2026; Ghafarollahi & Buehler, 2025), and mathematical reasoning (Lei et al., 2024; Zhang & Xiong, 2025; Zhou et al., 2025a). By assigning agents different roles and enabling information exchange among them, MAS can decompose complex tasks and leverage complementary capabilities during collaborative reasoning. Early systems typically adopt handcrafted communication structures, such as the sequential workflow in ChatDev (Qian et al., 2024) and role-based collaboration in CAMEL (Li et al., 2023) and MetaGPT (Hong et al., 2024). While effective, these predefined structures are generally fixed or manually designed and may not adapt well to varying task requirements. Recent studies therefore explore automated topology optimization to construct task-adaptive communication structures. Representative methods include G-Designer (Zhang et al., 2025), AGP (Li et al., 2025), and ARG-Designer (Li et al., 2026a), which dynamically optimize agent composition or connectivity through graph generation and pruning. Such advances shift MAS design from manually specified communication patterns toward automatically optimized collaboration structures. As these optimized topologies increasingly encode substantial computational investment and system design knowledge, protecting them from unauthorized inference becomes an important yet underexplored problem.

E.2 Adversarial Attacks against MAS

The growing adoption of LLM-based MAS has raised increasing concerns about their vulnerability to adversarial attacks (Yu et al., 2025; Kavathekar et al., 2026; Chen et al., 2026). Compared with standalone LLMs, MAS introduce additional attack surfaces through agent interactions, information propagation, and collaborative decision-making. Existing studies primarily target agent behaviors and

communication content, including prompt-based attacks (Lee et al., 2025; Shahroz et al., 2025; Arif et al., 2026), communication attacks (He et al., 2025b; Yan et al., 2026), and task disruption (Amayuelas et al., 2024). These attacks inject malicious instructions, manipulate inter-agent interactions, or exploit information propagation to compromise individual agents or overall system behavior.

Beyond such content- and behavior-oriented threats, the internal organization of MAS can itself become an attack target. CIA (Wu et al., 2026) reveals a distinct risk to topology confidentiality by reconstructing hidden communication topologies from semantic dependencies among observable reasoning traces under black-box access. Unlike attacks that primarily aim to manipulate task outputs, topology inference seeks to recover structural information about how agents interact. Such leakage can expose critical communication paths and proprietary system designs, creating both security and intellectual property risks. Despite these risks, defenses specifically targeting topology inference remain largely unexplored. This work fills this gap by introducing MIRAGE to protect MAS communication topologies against black-box inference.

E.3 PRIVACY & SECURITY PROTECTION FOR MAS

Alongside emerging attacks, increasing research attention has been devoted to improving the privacy and security of LLM-based MAS (Zhou et al., 2025b; Zhang et al., 2024; Raza et al., 2026; Miao et al., 2025; Liu et al., 2024; Zhan et al., 2025; Li et al., 2026b; Jha et al., 2026). Existing defenses mainly focus on content-level threats and malicious behaviors, protecting inter-agent collaboration through mechanisms such as attack detection, trust management, input isolation, and communication safeguards. For instance, BlindGuard (Miao et al., 2025) and GUARDIAN (Zhou et al., 2025b) introduce defensive mechanisms for securing agent interactions, while TRiSM (Raza et al., 2026) focuses on improving trust and security during multi-agent collaboration. Other defenses against adversarial instructions employ instructional prevention, delimiters, or detection mechanisms (Liu et al., 2024; Zhan et al., 2025) to prevent malicious inputs from influencing model behavior. However, these approaches primarily protect message content, agent behavior, or task execution rather than the communication topology itself. Protecting the content transmitted over an edge does not necessarily prevent an adversary from inferring whether that edge exists, particularly when structural relationships can be revealed through semantic dependencies among observable outputs. Consequently, the confidentiality of LLM-based MAS communication topologies remains largely overlooked by existing defenses. MIRAGE addresses this gap from a complementary perspective: rather than solely preventing malicious content from affecting agents, it conceals the genuine topology from black-box inference while preserving the original communication structure for task execution.

F LIMITATIONS & FUTURE DIRECTIONS

This section discusses the limitations of this work (▷ Sec. F.1) and outlines several promising directions for further exploration in future research (▷ Sec. F.2).

F.1 LIMITATIONS

Our experiments provide a comprehensive evaluation of MIRAGE across three representative topology optimization frameworks (G-Designer, AGP, and ARG-Designer) and four datasets (`MMLU`, `GSM8K`, `SVAMP`, and `HumanEval`) spanning general reasoning, mathematical reasoning, and code generation. The evaluation further covers diverse communication structures, multiple topology inference metrics, task utility, ablation studies, parameter sensitivity, and representative case studies, providing extensive empirical evidence for the effectiveness and generalizability of MIRAGE. Nevertheless, given the rapid evolution of LLM-based MAS, future systems may introduce new agent architectures, communication protocols, and application scenarios beyond the scope of our current evaluation, and the effectiveness of MIRAGE in such emerging settings remains to be further validated.

F.2 FUTURE DIRECTIONS

Several directions can further extend this work. *First*, MIRAGE can be explored in future LLM-based MAS architectures and communication protocols, particularly in systems involving dynamic topologies and long-horizon interactions. *Second*, integrating MIRAGE with complementary privacy and security mechanisms could provide more comprehensive protection for LLM-based multi-agent systems while maintaining their collaborative capabilities.