CIA: Inferring the Communication Topology from LLM-based Multi-Agent Systems

Yongxuan Wu[1,2], **Xixun Lin**[1,2*], **He Zhang**[3], **Nan Sun**[1,2], **Kun Wang**[4], **Chuan Zhou**[5], **Shirui Pan**[3], **Yanan Cao**[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Griffith University, Brisbane, Australia
[4] Nanyang Technological University, Singapore
[5] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
{wuyongxuan,linxixun}@iie.ac.cn

Abstract

LLM-based Multi-Agent Systems (MAS) have demonstrated remarkable capabilities in solving complex tasks. Central to MAS is the communication topology which governs how agents exchange information internally. Consequently, the security of communication topologies has attracted increasing attention. In this paper, we investigate a critical privacy risk: MAS communication topologies can be inferred under a restrictive black-box setting, exposing system vulnerabilities and posing significant intellectual property threats. To explore this risk, we propose **Communication Inference Attack (CIA)**, a novel attack that constructs new adversarial queries to induce intermediate agents' reasoning outputs and models their semantic correlations through the proposed global bias disentanglement and LLM-guided weak supervision. Extensive experiments on MAS with optimized communication topologies demonstrate the effectiveness of CIA, achieving an average AUC of 0.87 and a peak AUC of up to 0.99, thereby revealing the substantial privacy risk in MAS. The source code is available at https://github.com/aabbbcd/CIA.

1 Introduction

LLM-based agents have rapidly evolved into powerful intelligent systems, exhibiting human-like capabilities in cognition and reasoning (Shinn et al., 2023; Jin et al., 2023; Yang et al., 2024). To further scale these capabilities, recent research has shifted toward LLM-based multi-agent systems (MAS). By orchestrating the collaboration among multiple agents, MAS can tackle complex tasks that are beyond the reach of a single agent (Wang et al., 2024b; Li et al., 2024; Wang et al., 2025). As a result, MAS have demonstrated remarkable performance across various domains, including software engineering (He et al., 2025a), scientific discovery (Ghafarollahi and Buehler, 2025), and social simulation (Taillandier et al., 2025).

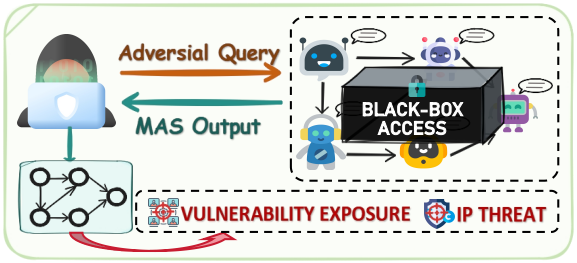


Figure 1: **Illustration of inferring the communication topology.** The adversary infers the communication topology under a black-box setting, resulting in a severe *Vulnerability Exposure* that enables adversaries to pinpoint critical agents for targeted attacks and *IP Threat* that leaks valuable proprietary assets of developers.

This advantage primarily stems from the optimized communication topology within MAS, which enables agents to exchange information and refine their decisions through collaboration or debate. Along with the rapid advancement of MAS, their security has attracted increasing attention (Wang et al., 2024a; Li et al., 2025a; Yan et al., 2026; Lin et al., 2025). In terms of adversarial attacks, existing studies mainly focus on inducing toxic outputs or misinformation spread among agents through various attack strategies (Lee and Tiwari, 2024; Yu et al., 2025c; He et al., 2025b). In this paper, we investigate a largely underexplored yet critical privacy risk: **Is the communication topology of MAS itself vulnerable to leakage?**

Compared with previous attacks, this is a stealthier privacy risk, as the adversary does not aim to disrupt task execution but seeks to infer the internal information of MAS solely through black-box queries. Moreover, as illustrated in Figure 1, once the communication topology is inferred, the consequences can be severe. First, it leads to *Vulnerability Exposure*. Revealing the communication topology of MAS exposes the systems' internal

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17917–17933
July 2-7, 2026

organization, allowing the adversary to identify critical or weak agents for targeted jailbreaking or instruction injection, thereby compromising MAS at low cost (Raza et al., 2026). Second, it poses a significant *IP Threat*. A highly optimized communication topology encapsulates substantial computational resources and expert knowledge, representing valuable proprietary assets (Zhang et al., 2025a; Li et al., 2025b). The leakage of this topology constitutes a direct infringement of IP, consequently undermining the developers' competitive advantage (Kong et al., 2025).

To explore this privacy risk, we propose **Communication Inference Attack (CIA)**, a novel attack for inferring the communication topology within MAS. CIA operates under a practical black-box setting, where the adversary is neither authorized to access nor alter any internal information of MAS. Instead, the adversary merely interacts with MAS by issuing queries and observing their final responses. CIA consists of two stages: *Reasoning Output Induction* and *Semantic Correlations Modeling*. In the first stage, CIA crafts adversarial queries to induce the final output to reveal the intermediate agents' reasoning outputs. In the second stage, we introduce global bias disentanglement and LLM-guided weak supervision to mitigate spurious correlations and enhance the topological information embedded in these reasoning outputs, and subsequently analyze their semantic correlations to infer the communication topology. Extensive evaluations demonstrate the effectiveness of CIA, revealing the substantial privacy risk in the communication topology of MAS. In summary, our contributions are as follows.

- We investigate a largely underexplored privacy risk in MAS: the vulnerability of their communication topologies to being inferred under a black-box setting, which poses significant IP threats and vulnerability exposure.
- We propose the CIA, a novel attack that first crafts adversarial queries to expose the reasoning outputs of intermediate agents and then models the semantic correlations of these outputs using global bias disentanglement and LLM-guided weak supervision to infer the confidential communication topology.
- We conduct experiments on MAS built using well-optimized communication topologies across multiple task scenarios. Experimental results show that the communication topology in MAS can be effectively inferred, with CIA achieving an average AUC of 0.87 and a peak AUC of up to 0.99.

2 Related Work

Topology Design for MAS. The communication topology is fundamental for the effectiveness of MAS, serving as the backbone of collective intelligence and joint reasoning (Cemri et al., 2025). Consequently, much work has focused on designing communication topologies for MAS (Liu et al., 2025). Early designs rely on handcrafted or heuristic patterns that lack the flexibility to adapt to diverse tasks (Hong et al., 2024; Li et al., 2023; Qian et al., 2024). To overcome this limitation, recent methods have introduced *Generative Optimization Strategies* to dynamically generate agent compositions or communication topologies tailored to specific tasks (Zhang et al., 2025a,b; Li et al., 2025b). These approaches not only achieve state-of-the-art (SOTA) performance but also reduce the resource costs of redundant communications in MAS.

Adversarial Attacks against MAS. Recent research on adversarial attacks against MAS has primarily focused on inducing toxic outputs or spreading misinformation (Yu et al., 2025a). Specifically, some methods study communication content-based attacks, such as task abandonment (Amayuelas et al., 2024), communication tampering (He et al., 2025b), or malicious prompt propagation (Lee and Tiwari, 2024; Yu et al., 2025c). Meanwhile, some approaches explore communication topology-based attacks by evaluating the resilience of different communication topologies to identify which topologies are more vulnerable to adversarial attacks (Huang et al., 2024; Yu et al., 2025b). However, the privacy risk of the communication topology itself remains largely unexplored. In this paper, we focus on this important risk and investigate whether the communication topology of MAS can be inferred in a black-box setting. inferring edges by exploiting correlations in prediction posteriors or gradient information.

3 Background

We introduce a basic framework of LLM-based MAS. Let $\mathcal{S} = (\mathcal{P}, \mathcal{G})$ represent MAS. Here, $\mathcal{P} = \{p_i\}_{i=1}^{n}$ denotes the set of agent profiles, each of which includes the agent's system prompt, callable tools, and other configuration details; $\mathcal{G} = (\mathcal{A}, \mathcal{E})$

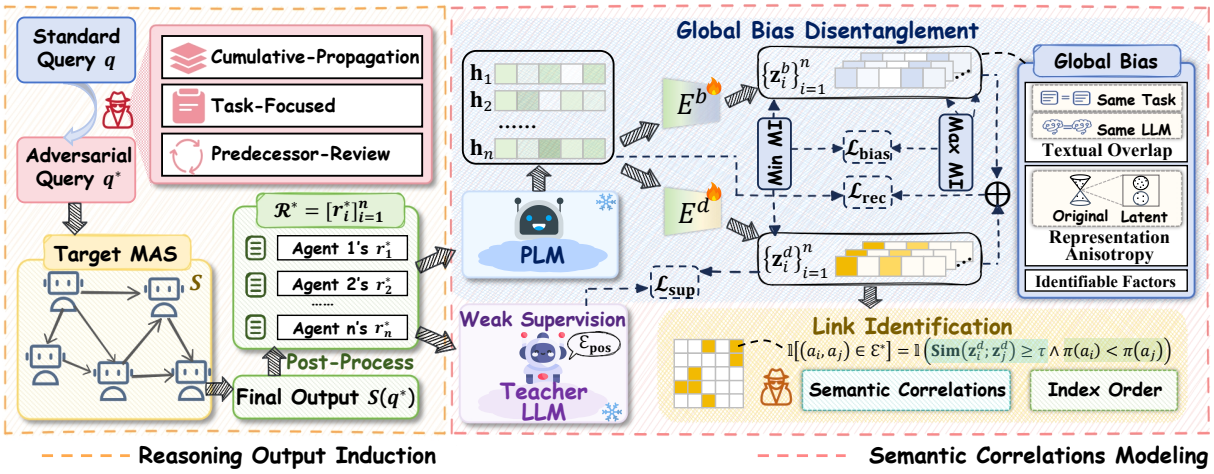


Figure 2: The overview of CIA. CIA first induces the final output that reveals the reasoning outputs of intermediate agents by crafting adversarial queries. CIA then infers the communication topology by modeling semantic correlations among these outputs using global bias disentanglement and LLM-guided weak supervision.

denotes the communication topology of $\mathcal{S}$, which is modeled as a directed acyclic graph (DAG) capturing the information flow for task completion (Zhang et al., 2025a; Li et al., 2026). $\mathcal{A} = \{a_i\}_{i=1}^n$ denotes the set of agents, each corresponding to an LLM, and $\mathcal{E} \subseteq \mathcal{A} \times \mathcal{A}$ denotes the set of directed edges. A directed edge $e_{j \to i} = (a_j, a_i) \in \mathcal{E}$ indicates that agent a_i is a designated recipient of information from agent a_j.

Based on this framework, for the given task with the corresponding query q, the i-th agent a_i generates its reasoning output r_i as follows:

$$r_i = \text{LLM}(p_i, q, \mathcal{O}_i). \tag{1}$$

$\mathcal{O}_i$ is the set of outputs generated by the predecessor agents of a_i, which can be defined as

$$\mathcal{O}_i = \{\, r_j \mid a_j \in \mathcal{N}_{\text{in}}(i) \,\}, \tag{2}$$

where $\mathcal{N}_{\text{in}}(i) = \{\, a_j \mid (a_j, a_i) \in \mathcal{E} \,\}$ denotes the set of predecessor agents of a_i. The final output of $\mathcal{S}$ is produced by the decision agent[1]:

$$r_n = \mathcal{S}(q) = \text{LLM}(p_n, q, \mathcal{O}_n). \tag{3}$$

From the above formulation, we obtain that the effective communication, governed explicitly by $\mathcal{G}$, is pivotal to the performance of $\mathcal{S}$, as it facilitates the efficient exchange and propagation of information among agents for task completion.

[1]This formulation follows a common abstraction in MAS, where the final output is generated by a decision agent via summarization or majority voting.

4 Research Problem

To investigate the privacy risk of communication topology leakage, we propose the Communication Inference Attack (CIA). Under our attack scenario, we introduce the system model, the adversary goal, and the adversary capabilities as follows.

System Model. We consider the MAS $\mathcal{S}$ designed to handle complex tasks, such as mathematical reasoning and code generation. In a standard usage scenario, a user provides a query q to the system, and $\mathcal{S}$ returns an output $\mathcal{S}(q)$ generated through collaborative interactions among agents.

Attack Goal. The adversary aims to infer the communication topology $\mathcal{G}$ through querying $\mathcal{S}$ and analyzing final outputs. This attack exposes MAS vulnerability by revealing its internal organization, enabling more targeted attacks on critical agents and threatens developers' IP.

Adversary Capabilities. The adversary operates under a practical black-box setting. This implies that the adversary can only interact with $\mathcal{S}$ through its external interface. The adversary has no access to the internal information of $\mathcal{S}$, such as reasoning traces, agent profiles, and system configurations.

5 Methodology

The intuition underlying CIA is that agents in $\mathcal{S}$ do not operate independently; instead, each agent's output is conditioned on the responses of its predecessors, resulting in stronger semantic dependencies between agents with direct topological connections than between those without such connections.

Under the black-box setting, CIA can only observe the final output r_n, as the internal information is inaccessible. Consequently, CIA first aims to induce $\mathcal{S}$ to reveal the reasoning outputs of intermediate agents. Here, intermediate agents refer to the agents in $\mathcal{S}$ excluding the decision agent. They participate in the intermediate reasoning process, but their outputs are not available. CIA then analyzes the semantic correlations between these outputs to infer the communication topology within $\mathcal{S}$.

Following this intuition, CIA naturally owns two stages. **1) Reasoning Output Induction**: CIA constructs adversarial queries to interact with the target MAS $\mathcal{S}$, inducing the final output that reveals reasoning outputs of intermediate agents. **2) Semantic Correlations Modeling**: CIA infers $\mathcal{G}$ by modeling the semantic correlations among these outputs. The overview of CIA is shown in Figure 2.

5.1 Reasoning Output Induction

This section presents a novel adversarial querying strategy for eliciting intermediate agents' reasoning from the final output. Concretely, the adversarial query imposes three specific constraints on each agent's output.

❶ **Cumulative-Propagation Constraint.** To ensure the final output contains the reasoning outputs of intermediate agents, we impose the cumulative-propagation constraint, requiring each agent to copy the historical record of its predecessors and append their reasoning outputs as the updated history. Through this cumulative recording process, the reasoning outputs are propagated through $\mathcal{G}$. The template for this constraint is as follows:

> **Cumulative-Propagation Constraint.**
>
> 1.**Goal**: Carry the history reasoning output.
> 2.**Action**: Copy the content of the previous agents' [PREVIOUS HISTORY] (if any) and append the previous agents' [REASONING OUTPUT] content.
> 3.**Format**: <Previous Agent [PREVIOUS HISTORY] Content> ||| <Previous Agent [REASONING OUTPUT] Content>.

❷ **Task-Focused Constraint.** The adversarial query inevitably introduces task-irrelevant information that can distract agents and cause deviations from their original reasoning trajectories. To mitigate this effect, we impose the task-focused constraint that requires each agent to focus exclusively on the task-relevant fields explicitly marked in the input and the reasoning outputs of its predecessors. The template for this constraint is as follows:

> **Task-Focused Constraint.**
>
> 1.**Goal**: Prevent reasoning deviation caused by task-irrelevant information.
> 2.**Action**: Exclusively extract and analyze the [TASK] and the previous agents' [REASONING OUTPUT] during your reasoning, disregarding any other information provided in the input.

❸ **Predecessor-Review Constraint.** To further strengthen the semantic correlations between the reasoning outputs of adjacent agents, we impose the predecessor-review constraint on each agent to review the predecessor agents' reasoning outputs before generating its own output. The template for this constraint is as follows:

> **Predecessor-Review Constraint.**
>
> 1.**Goal**: Strengthen the semantic association between predecessor and successor agents.
> 2.**Action**: Prior to your reasoning, explicitly review the previous agents' [REASONING OUTPUT] and incorporate them in your own [REASONING OUTPUT].

Guided by these three constraints, we use the adversarial query q^* to interact with $\mathcal{S}$ for generating $\mathcal{S}(q^*)$ that reveals the reasoning outputs of intermediate agents. Since $\mathcal{S}(q^*)$ is an unstructured text, we need to separate the outputs of each agent for the downstream semantic correlations modeling. Therefore, we post-process $\mathcal{S}(q^*)$ into a list, denoted as $\mathcal{R}^* = [r_i^*]_{i=1}^n$, where r_i^* corresponds to the reasoning output of i-th agent. This list $\mathcal{R}^*$ follows the order in which agents complete their reasoning task, implicitly reflecting the communication direction. Detailed procedures for this post-processing are provided in Appendix A.

5.2 Semantic Correlations Modeling

With the recovered agent outputs, we aim to infer $\mathcal{G}$ by modeling their semantic correlations in the following steps. First, we propose the global bias disentanglement to learn debiased representations for removing the spurious information in $\mathcal{R}^*$. Second, we design an LLM-guided weak supervision to refine these debiased representations, enhancing

their capability to learn the topological information of $\mathcal{G}$. Finally, we identify whether a link exists between agents by computing the similarity between their refined representations.

Global Bias Disentanglement (GBD). In fact, agents may still exhibit strong semantic similarity in their reasoning outputs even without explicit communication, rendering the semantic correlations among these recovered outputs can be highly spurious. Such spurious correlations stem from multiple sources. For instance, agents may share the same base LLM and operate on the same task, which naturally results in overlapping content and similar linguistic patterns (Bommasani et al., 2021). In addition, due to representation anisotropy (Godey et al., 2024), agents may produce semantically distinct outputs that nonetheless appear highly correlated in the embedding space. Beyond these identifiable factors, other unobservable sources may further exacerbate this issue (Chakrabarty, 2025).

We collectively refer to the sources that induce spurious correlations as **Global Bias**, since they represent the spurious information that is globally shared across agents' reasoning outputs. Such global bias can mislead the adversary into focusing on semantic signals that are unrelated to the communication topology, thereby inflating pairwise similarities among agents. As a consequence, many non-communicating agent pairs are falsely inferred as having communicated.

To mitigate the impact of global bias, we propose GBD to learn debiased representations. Specifically, we first employ a pretrained language model[2] f_θ to encode $\mathcal{R}^*$. For the i-th agent with its reasoning output r_i^*, f_θ produces an initial representation $\mathbf{h}_i$. We then project each $\mathbf{h}_i$ into two latent subspaces via two trainable encoders, E^d and E^b, representing the debiased encoder and the bias encoder, respectively:

$$\mathbf{z}_i^d = E^d(\mathbf{h}_i), \quad \mathbf{z}_i^b = E^b(\mathbf{h}_i), \tag{4}$$

where $\mathbf{z}_i^d$ and $\mathbf{z}_i^b$ denote the debiased and biased representations for r_i^*, respectively.

Since the global bias is shared across these reasoning outputs, we can maximize the mutual information among $\{\mathbf{z}_i^b\}_{i=1}^n$ to encourage E^b to effectively capture the global bias. Meanwhile, to prevent such global bias from influencing $\{\mathbf{z}_i^d\}_{i=1}^n$, we simultaneously minimize the mutual information between $\mathbf{z}_i^d$ and $\mathbf{z}_i^b$ for each agent. These two information-theoretic objectives are jointly optimized via the following loss:

$$\mathcal{L}_{\text{bias}} = -\mathcal{I}(\mathbf{z}_1^b; \ldots; \mathbf{z}_n^b) + \sum_{i=1}^{n} \mathcal{I}(\mathbf{z}_i^d; \mathbf{z}_i^b). \tag{5}$$

The computational details of Eq.(5) is given in Appendix B.

To prevent the encoded information from being lost during disentanglement (Bousmalis et al., 2016), we also introduce a reconstruction loss in GBD. Specifically, for each $\mathbf{h}_i$ with its disentangled $\mathbf{z}_i^d$ and $\mathbf{z}_i^b$, we concatenate them together and feed it into a decoder D to reconstruct $\mathbf{h}_i$:

$$\hat{\mathbf{h}}_i = D(\mathbf{z}_i^d \oplus \mathbf{z}_i^b). \tag{6}$$

The reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^{n} \left\| \mathbf{h}_i - \hat{\mathbf{h}}_i \right\|_2^2. \tag{7}$$

Finally, the overall training loss for GBD is

$$\mathcal{L}_{\text{GBD}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{bias}}. \tag{8}$$

LLM-guided Weak Supervision (LWS). At the above step, $\mathbf{z}_i^d$ is learned only from the textual information of r_i^*. We aim to enable $\mathbf{z}_i^d$ to capture the information at the structural level of $\mathcal{G}$, facilitating the more accurate communication inference. However, such information is not directly accessible. To this end, we leverage the information inferred by a teacher LLM[3] as weak supervision signals, distilling the structural knowledge into $\mathbf{z}_i^d$.

Given $\mathcal{R}^*$, the teacher LLM is prompted to infer the top-k communication edges with the highest confidence scores (The exact prompt template is provided in Appendix J.). We denote the edges inferred by this LLM as the positive set $\mathcal{E}_{\text{pos}}$, and sample negative pairs $\mathcal{E}_{\text{neg}}$ from the remaining agent pairs outside $\mathcal{E}_{\text{pos}}$. As the LLM-inferred edges may be noisy and the remaining pairs are not guaranteed to be true negatives, we adopt the trick of label smoothing (Dettmers et al., 2018) and define the weak supervision loss as

$$\begin{aligned} \mathcal{L}_{\text{LWS}} = &-\frac{1}{|\mathcal{E}_{\text{pos}}|} \sum_{(a_i,a_j)\in\mathcal{E}_{\text{pos}}} \mathcal{L}_{\text{pos}}(a_i, a_j) \\ &-\frac{1}{|\mathcal{E}_{\text{neg}}|} \sum_{(a_u,a_v)\in\mathcal{E}_{\text{neg}}} \mathcal{L}_{\text{neg}}(a_u, a_v), \end{aligned} \tag{9}$$

[2]We utilize the `all-MiniLM-L6-v2` in implementation.

[3]We utilize GPT-5 in our implementation.

where $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$ are the label-smoothed binary cross-entropy losses computed based on the similarity between the debiased representations for each positive and negative agent pair, respectively. Detailed formulations of $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$ are provided in Appendix C.

The trainable modules in CIA are E^d and E^b, and the final objective of CIA is to minimize the total loss:

$$\mathcal{L}_{\text{CIA}} = \mathcal{L}_{\text{GBD}} + \mathcal{L}_{\text{LWS}}. \tag{10}$$

Link Identification. After training, the communication topology $\mathcal{G}$ can be identified from the debiased representations. The existence of an edge between agents a_i and a_j is determined by the similarity between $\mathbf{z}_i^d$ and $\mathbf{z}_j^d$, while the edge direction is inferred according to their relative order in $\mathcal{R}^*$:

$$\mathbb{I}[(a_i, a_j) \in \mathcal{E}] = \mathbb{I}\Big(\text{Sim}(\mathbf{z}_i^d, \mathbf{z}_j^d) \geq \tau \wedge \pi(a_i) < \pi(a_j)\Big), \tag{11}$$

where $\text{Sim}(\cdot, \cdot)$ indicates a distance-based similarity function, τ is a threshold, and $\pi(a)$ denote the index of agent a's reasoning output in $\mathcal{R}^*$.

6 Experiments

6.1 Experiment Setups

MAS Frameworks. As introduced in Section 2, generative optimization strategies for communication topologies achieve SOTA performances. Unlike heuristic methods, these strategies often require substantial resources to design communication topologies carefully, making them more valuable targets for inferring. Accordingly, to evaluate CIA's performance, we construct communication topologies using three well-performing generative optimization strategies: G-Designer (Zhang et al., 2025a), AGP (Li et al., 2025b), and ARG-Designer (Li et al., 2026). More details about these strategies are provided in Appendix D.

Task Datasets. To provide tasks for generating diverse, task-driven communication topologies using the optimization strategies introduced above, we employ four datasets across three domains: ❶ General Reasoning: MMLU (Hendrycks et al., 2021); ❷ Mathematical Reasoning: GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021); and ❸ Code Generation: HumanEval (Chen et al., 2021). For each dataset, we select 100 tasks for evaluation. More details about the datasets are provided in Appendix E.

Baselines. We select two closed-source LLMs (GPT-5 and Gemini-2.5-Pro) and two open-source LLMs (Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.2) as our baseline attacks for inferring the communication topology. Specifically, we prompt them to assign confidence scores to all agent pairs for inferring the communication. We then apply a threshold of 0.5 to these scores to determine the predicted edges for evaluation. The exact prompt template is provided in Appendix K.

Metrics. We evaluate the performance of topology inference by measuring the prediction accuracy of all possible edges within MAS. We report Area Under the ROC Curve (AUC) (Hanley and McNeil, 1982), Accuracy (ACC) and F1-score (F1) (Željko Đ. Vujovic, 2021).

6.2 Inference Attack Performance

Table 1 demonstrates the communication inference results of MAS constructed by three generative optimization strategies across four datasets. The best performance is in boldface. From Table 1, we have the following observations:

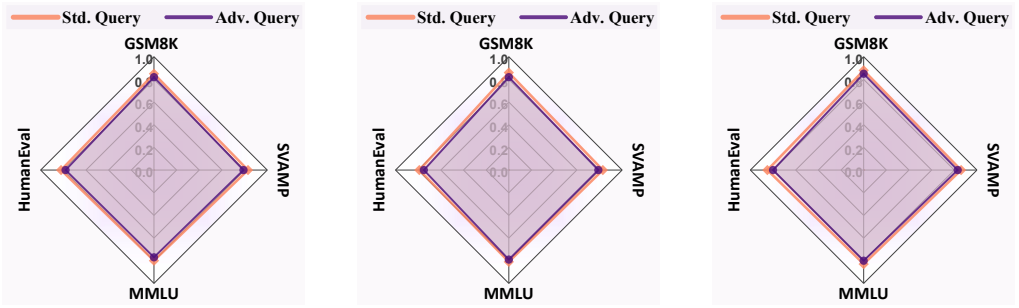
❶ The communication topology can be inferred. CIA exhibits superior performance in communication inference, with an AUC exceeding 0.75 in all cases and surpassing 0.80 in most experiments, peaking at 0.99, revealing the critical privacy risk in the communication topology of MAS.

❷ A simpler communication topology is more susceptible to being inferred. As shown in Table 2, ARG-Designer constructs MAS for GSM8K and SVAMP with significantly fewer average nodes and edges compared to other MAS, and our CIA achieves an AUC close to 1.0 in these cases. While having fewer nodes and edges results in lower resource consumption, it increases the risk of the communication topology leakage.

❸ CIA significantly outperforms all LLM baselines. Among the baselines, closed-source models generally exhibit stronger reasoning capabilities compared to open-source models. However, all LLMs tend to assign lower confidence scores to the communication between agents, failing to effectively distinguish whether there is communication between them.

Dataset	Method	MMLU			GSM8K			SVAMP			HumanEval		
		AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1
G-Designer	GPT-5	0.5833	0.8033	0.3084	0.6274	0.5887	0.4555	0.6272	0.7876	0.6815	0.6201	0.5211	0.3204
	Gemini-2.5-Pro	0.6869	0.7900	0.3697	0.6152	0.5728	0.4545	0.6300	0.7600	0.7059	0.6318	0.5541	0.3899
	Llama-3.1-8B-Instruct	0.5995	0.7157	0.4156	0.5284	0.5348	0.4689	0.4567	0.6252	0.4361	0.4922	0.4626	0.2863
	Mistral-7B-Instruct-v0.2	0.5192	0.7555	0.2889	0.4105	0.4260	0.2866	0.4977	0.6704	0.4351	0.4893	0.4430	0.2811
	CIA (Ours)	**0.8324**	**0.8147**	**0.7761**	**0.8585**	**0.8382**	**0.7744**	**0.8561**	**0.8083**	**0.7682**	**0.7594**	**0.7362**	**0.6566**
AGP	GPT-5	0.6577	0.6974	0.5472	0.6233	0.5972	0.5357	0.6199	0.7940	0.6330	0.4433	0.6661	0.2689
	Gemini-2.5-Pro	0.6324	0.5935	0.5217	0.6177	0.5733	0.5174	0.6087	0.7733	0.6241	0.4033	0.6122	0.2158
	Llama-3.1-8B-Instruct	0.4812	0.6581	0.4786	0.4265	0.5190	0.5100	0.4355	0.6315	0.5396	0.3961	0.6450	0.2500
	Mistral-7B-Instruct-v0.2	0.5076	0.6579	0.4246	0.4867	0.6500	0.2866	0.4939	0.6588	0.4475	0.5000	0.6722	0.2857
	CIA (Ours)	**0.8411**	**0.7942**	**0.7567**	**0.8804**	**0.8204**	**0.7866**	**0.8979**	**0.8380**	**0.7923**	**0.8100**	**0.7777**	**0.6759**
ARG-Designer	GPT-5	0.5879	0.7545	0.4094	0.6984	0.7672	0.5113	0.6240	0.7163	0.4034	0.6092	0.7294	0.5755
	Gemini-2.5-Pro	0.6888	0.7795	0.4482	0.7475	0.7977	0.5743	0.6197	0.7313	0.4097	0.6006	0.7601	0.5531
	Llama-3.1-8B-Instruct	0.4912	0.6511	0.3899	0.5123	0.7345	0.6435	0.5349	0.6277	0.4352	0.3443	0.6204	0.3445
	Mistral-7B-Instruct-v0.2	0.4539	0.6465	0.2649	0.5713	0.6697	0.4924	0.3112	0.5347	0.1235	0.2936	0.5875	0.1653
	CIA (Ours)	**0.8227**	**0.7931**	**0.7458**	**0.9873**	**0.9035**	**0.8330**	**0.9761**	**0.9106**	**0.8632**	**0.8699**	**0.8126**	**0.7153**

Table 1: Comparison of inference attack performance between CIA and baselines.



(a) G-Designer. (b) AGP. (c) ARG-Designer.

Figure 3: Comparison of MAS utility (measured by Accuracy) between **Std.Query** and **Adv.Query**.

Dataset	G-Designer		AGP		ARG-Designer	
	N_{avg}	E_{avg}	N_{avg}	E_{avg}	N_{avg}	E_{avg}
MMLU	7.0	8.99	6.0	10.87	5.42	7.84
GSM8K	5.0	8.19	5.0	8.45	3.07	3.14
SVAMP	5.0	8.15	5.0	8.41	3.05	3.10
HumanEval	6.0	11.38	6.0	11.54	4.24	5.49

Table 2: Statistical details of generated communication topologies. We report the average number of nodes (N_{avg}) and edges (E_{avg}) for each setting.

6.3 Effectiveness of Adversarial Query

In this section, we evaluate our adversarial query strategy (Section 5.1) by studying two primary factors: (1) **the fidelity of raw reasoning output recovery**, and (2) **the impact of adversarial query on MAS task performance**. The latter ensures that our strategy does not degrade system functionality, which would otherwise render the inferred topologies meaningless.

For the first factor, we use Recall (Rec) to measure the proportion of recovered agent reasoning outputs based on high semantic similarity to the ground truth, and use ROUGE-L (R-L) to evaluate the lexical precision and structural fidelity of the recovered outputs. As shown in Table 3, our reasoning outputs induction achieves robust recovery performance across various scenarios. Notably, the effectiveness is more pronounced in MAS generated by ARG-Designer, which have simpler topologies, thus minimizing the information loss caused by multi-source aggregation during the reasoning process. thus minimizing the information loss caused by multi-source aggregation during the reasoning process.

Dataset	G-Designer		AGP		ARG-Designer	
	Rec	R-L	Rec	R-L	Rec	R-L
MMLU	0.90	0.89	0.89	0.88	0.93	0.93
GSM8K	0.92	0.91	0.91	0.91	0.96	0.95
SVAMP	0.92	0.92	0.93	0.92	0.95	0.94
HumanEval	0.87	0.87	0.88	0.87	0.93	0.92

Table 3: Output recovery effectiveness via Adv. Query.

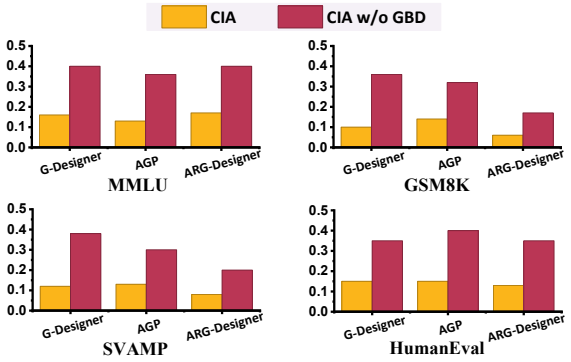


Figure 4: Impact of GBD on FPR.

G-Designer	MMLU	GSM8K	SVAMP	HumanEval
CIA w/o GBD	0.5264	0.5391	0.5308	0.5115
CIA	0.8324	0.8585	0.8561	0.7594
AGP	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA w/o GBD	0.5274	0.5683	0.5857	0.5252
CIA	0.8411	0.8804	0.8979	0.7480
ARG-Designer	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA w/o GBD	0.5333	0.6268	0.6294	0.6128
CIA	0.8227	0.9873	0.9761	0.8699

Table 4: Impact of GBD on attack performance (AUC).

For the second factor, we compare the task completion accuracy between standard query (Std.Query) and our adversarial query (Adv.Query). As illustrated in Figure 3, the performance under the Adv.Query is nearly identical to that of Std.Query across all settings. This shows that our strategy does not degrade system performance, confirming that the inferred communication accurately reflects how MAS solve complex reasoning problems. Furthermore, by preserving collaborative integrity, CIA remains stealthy and indistinguishable from benign usage.

6.4 Effectiveness of GBD

In this section, we evaluate the effectiveness of GBD by comparing the False Positive Rate (FPR) and communication inference performance between CIA and CIA w/o GBD (the model variant without GBD). CIA outperforms CIA w/o GBD. As illustrated in Figure 4, by eliminating global bias, CIA substantially reduces false positives, achieving at least a 50% reduction in FPR across all settings. Moreover, as reported in Table 4, removing global bias mitigates the spurious correlations among agents' reasoning outputs, leading to a significant improvement in communication inference.

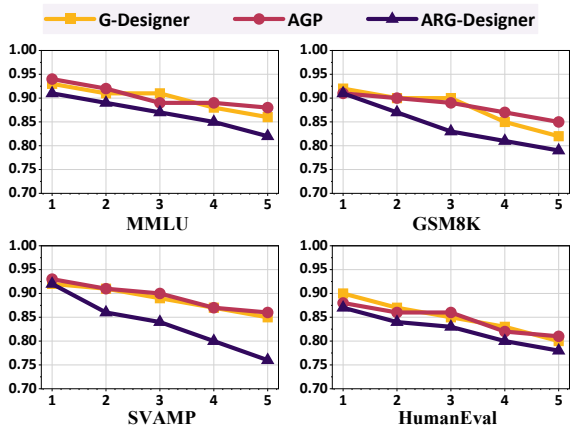


Figure 5: Precision of Top-k high-confidence edges.

G-Designer	MMLU	GSM8K	SVAMP	HumanEval
CIA w/o LWS	0.7856	0.8042	0.7852	0.7348
CIA	0.8324	0.8585	0.8561	0.7594
AGP	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA w/o LWS	0.7960	0.8243	0.8471	0.7061
CIA	0.8411	0.8804	0.8979	0.7480
ARG-Designer	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA w/o LWS	0.7671	0.9012	0.8992	0.7724
CIA	0.8227	0.9873	0.9761	0.8699

Table 5: Impact of LWS on attack performance (AUC).

6.5 Effectiveness of LWS

In this section, we evaluate the effectiveness of LLM-guided Weak Supervision (LWS). Since LLMs struggle with direct full-topology inference as shown in Table 1, we first assess the precision of Top-k high-confidence edges to verify the reliability of these supervision signals. We then conduct an ablation study to compare CIA with CIA w/o LWS (the model variant without LWS). Figure 5 demonstrates that the LLM performs well in the precision evaluation of Top-k high-confidence edges, particularly where $k \leq 3$, indicating that these weak supervision signals are reliable for inference. Consequently, as shown in Table 5, LWS improves AUC across all settings, validating its effectiveness in refining the debiased representations and enhancing the inference performance.

7 Conclusion

This paper investigates the privacy risk of MAS communication topologies being inferred, which poses significant security and IP threats. We propose a restrictive black-box attack, CIA, which operates in two stages: first, it constructs adver-

sarial queries to reveal all agent reasoning outputs; second, it infers the communication topology by analyzing the semantic correlations among agents using global bias disentanglement and LLM-guided weak supervision. Extensive experiments show that CIA can effectively infer communication topologies, highlighting the inherent privacy risk of MAS communication.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62402491, No.U2336202, No.62472416) and the China Postdoctoral Science Foundation (No.2025M771524).

Limitations

CIA employs a recursive estimation of Total Correlation (TC) to optimize the information-theoretic objectives in Eq. (5), as detailed in Appendix B. However, accurately estimating multivariate mutual information among high-dimensional vectors remains highly challenging, leaving room for improvement in our approximation strategy. Moreover, the current LLM-guided weak supervision in CIA captures only first-order topological information; exploiting higher-order topological patterns to further strengthen the attack remains an open research direction.

References

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. 2024. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 6929–6948. Association for Computational Linguistics.

Ke Bai, Pengyu Cheng, Weituo Hao, Ricardo Henao, and Larry Carin. 2023. Estimating total correlation with mutual information estimators. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, Proceedings of Machine Learning Research, pages 2147–2164. PMLR.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. 2018. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Proceedings of Machine Learning Research, pages 530–539. PMLR.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, and 34 others. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 343–351.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya G. Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Why do multi-agent LLM systems fail? *CoRR*, abs/2503.13657.

Pradipta Kishore Chakrabarty. 2025. Causal inference in agentic ai: Bridging explainability and dynamic decision making. *International Journal of Science and Research (IJSR)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

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

Nathan Godey, Éric Villemonte de la Clergerie, and Benoît Sagot. 2024. Anisotropy is inherent to self-attention in transformers. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 -*

Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024, pages 35–48. Association for Computational Linguistics.

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Junda He, Christoph Treude, and David Lo. 2025a. Llm-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Trans. Softw. Eng. Methodol.*, 34(5):124:1–124:30.

Pengfei He, Yuping Lin, Shen Dong, Han Xu, Yue Xing, and Hui Liu. 2025b. Red-teaming LLM multi-agent systems via communication attacks. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 6726–6747. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jen-tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Maarten Sap, and Michael R. Lyu. 2024. On the resilience of multi-agent systems with malicious agents. *CoRR*, abs/2408.00989.

Ye Jin, Xiaoxi Shen, Huiling Peng, Xiaoan Liu, Jingli Qin, Jiayang Li, Jintao Xie, Peizhong Gao, Guyue Zhou, and Jiangtao Gong. 2023. Surrealdriver: Designing generative driver agent simulation framework in urban contexts based on large language model. *CoRR*, abs/2309.13193.

Dezhang Kong, Shi Lin, Zhenhua Xu, Zhebo Wang, Minghao Li, Yufeng Li, Yilun Zhang, Hujin Peng, Zeyang Sha, Yuyuan Li, Changting Lin, Xun Wang, Xuan Liu, Ningyu Zhang, Chaochao Chen, Muhammad Khurram Khan, and Meng Han. 2025. A survey of llm-driven AI agent communication: Protocols, security risks, and defense countermeasures. *CoRR*, abs/2506.19676.

Donghyun Lee and Mo Tiwari. 2024. Prompt infection: Llm-to-llm prompt injection within multi-agent systems. *CoRR*, abs/2410.07283.

Ang Li, Yin Zhou, Vethavikashini Chithrra Raghuram, Tom Goldstein, and Micah Goldblum. 2025a. Commercial LLM agents are already vulnerable to simple yet dangerous attacks. *CoRR*, abs/2502.08586.

Boyi Li, Zhonghan Zhao, Der-Horng Lee, and Gaoang Wang. 2025b. Adaptive graph pruning for multi-agent communication. In *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025)*, Frontiers in Artificial Intelligence and Applications, pages 4305–4312. IOS Press.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More agents is all you need. *Trans. Mach. Learn. Res.*, 2024.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. 2026. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 23142–23150. AAAI Press.

Xixun Lin, Yucheng Ning, Jingwen Zhang, Yan Dong, Yilong Liu, Yongxuan Wu, Xiaohua Qi, Nan Sun, Yanmin Shang, Pengfei Cao, Lixin Zou, Xu Chen, Chuan Zhou, Jia Wu, Shirui Pan, Bin Wang, Yanan Cao, Kai Chen, Songlin Hu, and Li Guo. 2025. Llm-based agents suffer from hallucinations: A survey of taxonomy, methods, and directions. *CoRR*, abs/2509.18970.

Yixin Liu, Guibin Zhang, Kun Wang, Shiyuan Li, and Shirui Pan. 2025. Graph-augmented large language model agents: Current progress and future prospects. *CoRR*, abs/2507.21407.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. 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-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Chatdev: Communicative

agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15174–15186. Association for Computational Linguistics.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. 2026. Trism for agentic AI: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *AI Open*, 7:71–95.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Patrick Taillandier, Jean-Daniel Zucker, Arnaud Grignard, Benoît Gaudou, Nghi Quang Huynh, and Alexis Drogoul. 2025. Integrating LLM in agent-based social simulation: Opportunities and challenges. *CoRR*, abs/2507.19364.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Qian Wang, Tianyu Wang, Zhenheng Tang, Qinbin Li, Nuo Chen, Jingsheng Liang, and Bingsheng He. 2025. Megaagent: A large-scale autonomous llm-based multi-agent system without predefined sops. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 4998–5036. Association for Computational Linguistics.

Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. 2024a. Badagent: Inserting and activating backdoor attacks in LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9811–9827. Association for Computational Linguistics.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024b. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 257–279. Association for Computational Linguistics.

Michael Satosi Watanabe. 1960. Information theoretical analysis of multivariate correlation. *IBM J. Res. Dev.*, 4(1):66–82.

Bingyu Yan, Xiaoming Zhang, Ziyi Zhou, Chaozhuo Li, Ruilin Zeng, Yirui Qi, Tianbo Wang, and Litian Zhang. 2026. Attack the messages, not the agents: A multi-round adaptive stealthy tampering framework for LLM-MAS. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 29784–29792. AAAI Press.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pang, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, Bo An, and Qingsong Wen. 2025a. 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, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pages 6216–6226. ACM.

Miao Yu, Shilong Wang, Guibin Zhang, Junyuan Mao, Chenlong Yin, Qijiong Liu, Kun Wang, Qingsong Wen, and Yang Wang. 2025b. Netsafe: Exploring the topological safety of multi-agent system. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 2905–2938. Association for Computational Linguistics.

Weichen Yu, Kai Hu, Tianyu Pang, Chao Du, Min Lin, and Matt Fredrikson. 2025c. Infecting LLM agents via generalizable adversarial attack. In *Red Teaming GenAI: What Can We Learn from Adversaries?*

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2025a. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025b. Aflow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yangze Zhou, Gitta Kutyniok, and Bruno Ribeiro. 2022. OOD link prediction generalization capabilities of message-passing gnns in larger test graphs. In *Advances in Neural Information Processing Systems 35:*

Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.

Željko Đ. Vujovic. 2021. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6).

A Details of Post-processing Procedures

This section details the post-processing procedures applied to $\mathcal{S}(q^*)$. As illustrated in Figure 6, $\mathcal{S}(q^*)$ consists of two sections: [PREVIOUS HISTORY] and [REASONING OUTPUT]. To format $\mathcal{S}(q^*)$ into the structured list $\mathcal{R}^*$ for downstream communication inference, firstly, we partition the [PREVIOUS HISTORY] section of $\mathcal{S}(q^*)$ using the delimiter "|||" to extract reasoning outputs from intermediate agents. Next, since a predecessor's output may be inherited by multiple subsequent agents, we apply backward deduplication to the partitioned results. Finally, since the [PREVIOUS HISTORY] section in $\mathcal{S}(q^*)$ doesn't record the output of the decision agent, we append the content in the [REASONING OUTPUT] section of $\mathcal{S}(q^*)$.

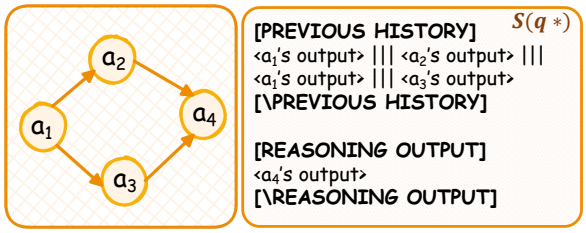


Figure 6: An illustrative example of $\mathcal{S}(q^*)$.

B Computational Details of Eq.(5)

In this section, we detail the computation process of Eq.(5). The equation consists of two terms: (i) the negative multivariate mutual information among all outputs' bias representations: $-\mathcal{I}(\mathbf{z}_1^b;\ldots;\mathbf{z}_n^b)$, and (ii) the sum of mutual information between the debiased and bias representations for each agent's output: $\sum_{i=1}^{n}\mathcal{I}(\mathbf{z}_i^d;\mathbf{z}_i^b)$.

For the first term, we introduce Total Correlation to estimate the multivariate mutual information. First, we provide the definitions of mutual information (MI) (Belghazi et al., 2018) and Total Correlation (TC) (Watanabe, 1960).

Given two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, their MI is defined as

$$\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[\log\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}\right] \tag{12}$$

For multi-variable scenarios, TC is defined as

$$\begin{aligned}\mathcal{TC}(\boldsymbol{X}) &= \mathcal{TC}(\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_n)\\ &= \mathbb{E}_{p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)}\left[\log\frac{p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)}{p(\boldsymbol{x}_1)\ldots p(\boldsymbol{x}_n)}\right].\end{aligned} \tag{13}$$

Based on the above definitions, the connection between TC and MI is described by the following theorem:

Theorem 1 (Bai et al., 2023). *Let* $\boldsymbol{X} = (\boldsymbol{x_1},\boldsymbol{x_2},\ldots,\boldsymbol{x_n})$ *be a group of random variables. Suppose set* $\mathcal{A} = \{i_1,i_2,\ldots,i_k\} \subseteq \{1,2,\ldots,n\}$ *is an index subgroup.* $\bar{\mathcal{A}} = \{i : i \notin \mathcal{A}\}$ *is the complementary set of* $\mathcal{A}$. *Denote* $\boldsymbol{X}_{\mathcal{A}} = (x_{i_1},x_{i_2},\ldots,x_{i_k})$ *as the selected variables from* $\boldsymbol{X}$ *with the indexes in* $\mathcal{A}$. *Then we have* $\mathcal{TC}(\boldsymbol{X}) = \mathcal{TC}(\boldsymbol{X}_{\mathcal{A}}) + \mathcal{TC}(\boldsymbol{X}_{\bar{\mathcal{A}}}) + \mathcal{I}(\boldsymbol{X}_{\mathcal{A}};\boldsymbol{X}_{\bar{\mathcal{A}}})$.

Theorem 1 reveals that TC can be equivalently decomposed into the internal correlations within subgroups and the MI between these subgroups. Based on this property, we can recursively represent the TC of the subgroups in terms of MI terms for lower-level subgroups. Consequently, by iteratively partitioning the set into a preceding subset $\boldsymbol{X}_{1:i}$ and the current variable $\boldsymbol{x}_{i+1}$, TC can be formulated as a summation of progressive MI terms:

$$\mathcal{TC}(\boldsymbol{X}) = \sum_{i=1}^{n-1}\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1}), \tag{14}$$

Based on this, our multivariate mutual information can be reformulated as

$$\mathcal{I}(\mathbf{z}_1^b;\ldots;\mathbf{z}_n^b) = \mathcal{TC}(\mathbf{Z}^b) = \sum_{i=1}^{n-1}\mathcal{I}(\mathbf{Z}_{1:i}^b;\mathbf{z}_{i+1}^b), \tag{15}$$

where $\mathbf{Z}^b = (\mathbf{z}_1^b,\ldots,\mathbf{z}_n^b)$, and $\mathbf{Z}_{1:i}^b = (\mathbf{z}_1^b,\ldots,\mathbf{z}_i^b)$ denote a subset of variables with indexes from 1 to i.

To estimate each MI term, we perturb each task multiple times to elicit diverse responses and use the collected outputs to form an empirical approximation of each agent's output distribution. We then apply the InfoNCE (van den Oord et al., 2018) to estimate the MI terms in the summation:

$$\begin{aligned}\mathcal{I}(\mathbf{Z}_{1:i}^b;\mathbf{z}_{i+1}^b) = \mathbb{E}\Bigg[\frac{1}{M}\sum_{p=1}^{M}\Big(\phi(\mathbf{Z}_{1:i,p}^b,\mathbf{z}_{i+1,p}^b)\\ -\log\Big(\frac{1}{M}\sum_{q=1}^{M}\exp\big(\phi(\mathbf{Z}_{1:i,p}^b,\mathbf{z}_{i+1,q}^b)\big)\Big)\Big)\Bigg],\end{aligned} \tag{16}$$

For the second term in Eq.(5), we also use InfoNCE to estimate the MI between the debiased and bias representations for each agent's output:

$$\mathcal{I}(\mathbf{z}_i^d; \mathbf{z}_i^b) = \mathbb{E}\Bigg[\frac{1}{M}\sum_{p=1}^{M}\Big(\phi(\mathbf{z}_{i,p}^d, \mathbf{z}_{i,p}^b) - \log\Big(\frac{1}{M}\sum_{q=1}^{M}\exp\big(\phi(\mathbf{z}_{i,p}^d, \mathbf{z}_{i,q}^b)\big)\Big)\Big)\Bigg]. \tag{17}$$

C Formulations of $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$

Here we first give the formulation of $\mathcal{L}_{\text{pos}}$:

$$\mathcal{L}_{\text{pos}}(a_i, a_j) = (1-\alpha)\cdot\log(\text{Sim}(\mathbf{z}_i^d, \mathbf{z}_j^d)) + \alpha\cdot\log(1-\text{Sim}(\mathbf{z}_i^d, \mathbf{z}_j^d)), \tag{18}$$

where α is the label-smoothing coefficient, $\text{Sim}(\cdot,\cdot)$ is a distance-based similarity function, and $\mathbf{z}_i^d$ and $\mathbf{z}_j^d$ denote the debiased representations corresponding to the outputs of agents a_i and a_j. Accordingly, the formulation of $\mathcal{L}_{\text{neg}}$ is

$$\mathcal{L}_{\text{neg}}(a_u, a_v) = (1-\alpha)\cdot\log(1-\text{Sim}(\mathbf{z}_u^d, \mathbf{z}_v^d)) + \alpha\cdot\log(\text{Sim}(\mathbf{z}_u^d, \mathbf{z}_v^d)). \tag{19}$$

D Generative Optimization Strategies

The generative optimization strategies for the communication topology used in our experiments are introduced below.

- G-Designer (Zhang et al., 2025a) is a topology optimization framework that learns effective multi-agent communication topologies by modeling agent interactions as a graph and optimizing connectivity to improve collaborative reasoning performance.
- AGP (Li et al., 2025b) proposes an adaptive graph pruning strategy that iteratively removes redundant or ineffective communication links, resulting in more efficient and task-relevant multi-agent interaction topologies.
- ARG-Designer (Li et al., 2026) reframes multi-agent system design as conditional autoregressive graph generation. By jointly optimizing agent composition and topology, it constructs customized topologies from scratch to enable task-adaptive coordination.

E Task Datasets

The task datasets used in our experiments are introduced below.

- MMLU (Hendrycks et al., 2021) is a benchmark designed to evaluate general reasoning and knowledge understanding across diverse subject domains, covering a wide range of factual, logical, and conceptual questions.
- GSM8K (Cobbe et al., 2021) focuses on complex, multi-step mathematical reasoning, requiring models to perform precise arithmetic operations and logical deductions to solve diverse, grade-school–level word problems.
- SVAMP (Patel et al., 2021) targets robustness in mathematical reasoning by introducing systematic structural and linguistic variations to math problems, testing whether models truly understand complex problem semantics rather than relying on spurious surface cues.
- HumanEval (Chen et al., 2021) is a code generation benchmark that assesses a model's ability to synthesize correct and executable programs from natural language specifications.

F Model Variant for Disentanglement

In this section, we introduce a model variant, termed CIA-Sub, which approaches the distanglement of global bias in a different way. Instead of using two separate encoders to learn the debiased representations and the bias representations, CIA-Sub uses a single encoder to obtain the bias representations, while the debiased representations are defined by subtracting the bias representations from the initial representations:

$$\mathbf{z}_i^b = E^b(\mathbf{h}_i), \quad \mathbf{z}_i^d = \mathbf{h}_i - \mathbf{z}_i^b. \tag{20}$$

All loss components for CIA-Sub remain unchanged from those of CIA.

As shown in Table 6, CIA performs better than CIA-Sub. We suppose the reason is that in CIA-Sub, $\mathbf{z}_i^d$ is entirely dependent on the quality of $\mathbf{z}_i^b$, while in CIA, using two separate encoders allows the debiased representations to be explicitly refined to capture useful information relevant to the communication.

G-Designer	MMLU	GSM8K	SVAMP	HumanEval
CIA-Sub	0.7432	0.7689	0.7455	0.6509
CIA	0.8324	0.8585	0.8561	0.7594
AGP	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA-Sub	0.7181	0.7625	0.7964	0.6622
CIA	0.8411	0.8804	0.8979	0.7480
ARG-Designer	**MMLU**	**GSM8K**	**SVAMP**	**HumanEval**
CIA-Sub	0.7454	0.8654	0.8294	0.7669
CIA	0.8227	0.9873	0.9761	0.8699

Table 6: AUC comparison between CIA-Sub and CIA.

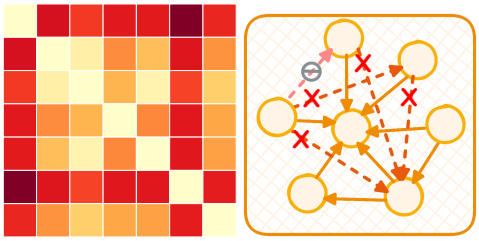
G Case studies

In this section, we present case studies for communication inference. Specifically, we visualize the similarity matrices and the inferred communication topologies produced by CIA w/o GBD and CIA, and compare them against the ground-truth adjacency matrix and communication topology. Notably, since the similarity matrix is symmetric, we symmetrize the ground-truth adjacency matrix for a more intuitive comparison.

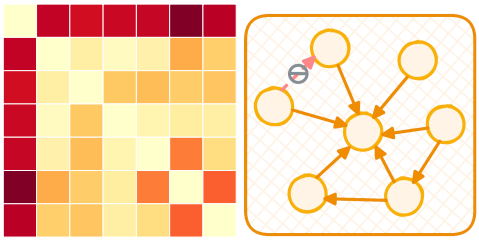
Figures 7 to 9 present three case studies, where the communication topologies are generated by G-Designer, AGP, and ARG-Designer on MMLU, respectively. These case studies visualize the spurious correlations induced by global bias and their impact on communication inference, thereby demonstrating the effectiveness of our GBD.

H Implementation Details

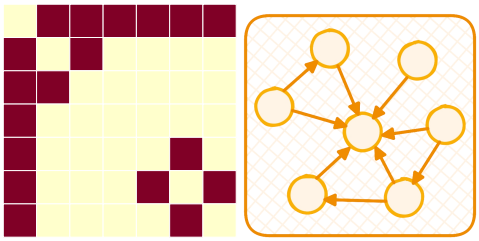
For our implementation, we utilize all-MiniLM-L6-v2 as the pretrained language model f_θ to generate the initial representations. The dimensions of both the debiased and biased representations in Global Bias Disentanglement (GBD) are set to 768, which is twice the native output dimension of f_θ. We employ GPT-5 as the teacher LLM to provide weak supervision signals in LLM-guided Weak Supervision (LWS). The threshold τ in Eq.(11) is set to 0.5, following common practice (Zhou et al., 2022). Additionally, the label-smoothing coefficient α in Eq.(18) and Eq.(19) is set to 0.1, consistent with the practice in Dettmers et al. (2018).



(a) CIA w/o GBD.

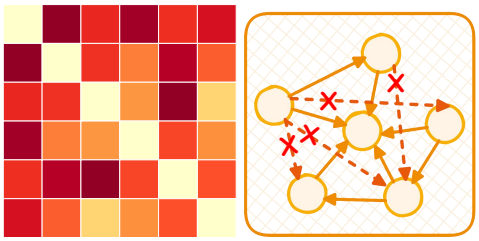


(b) CIA.

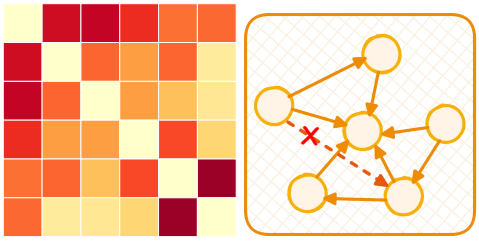


(c) Ground-truth.

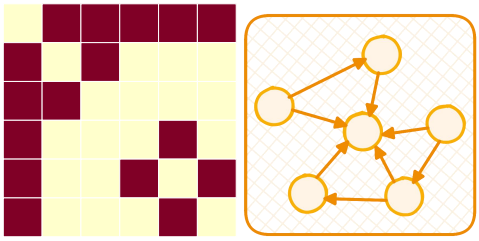
Figure 7: A case study of the communication topology generated by G-Designer on MMLU.



(a) CIA w/o GBD.

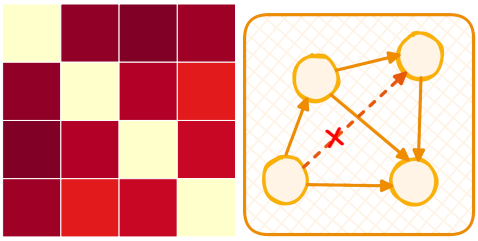


(b) CIA.

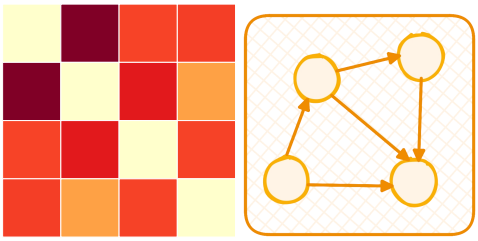


(c) Ground-truth.

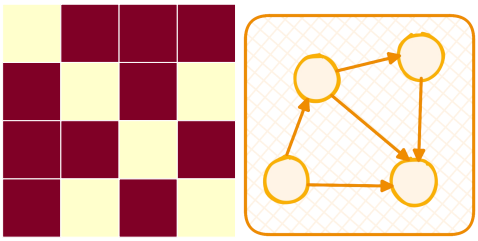
Figure 8: A case study of the communication topology generated by AGP on MMLU.



(a) CIA w/o GBD.



(b) CIA.



(c) Ground-truth.

Figure 9: A case study of the communication topology generated by ARG-Dsigner on MMLU.

I Hyperparameter Analysis

We conduct a grid search to select the optimal hyperparameter values. Specifically, we focus on tuning two primary parameters: the learning rate lr and k, the number of the highest confidence scores edges returned by the teacher LLM in LWS. The search intervals for these parameters are $\{1e-4, 5e-4, 1e-3, 5e-3, 1e-2\}$ and $\{1, 2, 3, 4, 5\}$, respectively. We conduct these hyperparameter experiments on MAS generated by G-Designer. The corresponding results are illustrated in Figure 10. From it, we have the following conclusions:

❶ CIA performs best at a learning rate of 1e-3, so appropriately increasing the learning rate can improve the CIA's inference quality. A smaller learning rate would result in slow convergence and inadequate learning, while a larger learning rate may cause gradient oscillations, slightly degrading performance.

❷ CIA achieves the best performance when $k = 3$. A smaller k makes the debiased representations difficult to capture sufficient topological information, while for a large k, the teacher LLM's precision decreases more noticeably, as shown in Figure 5, introducing more incorrect edges, which brings noise and misleads $\mathbf{z}_i^d$ to learn topological information deviating from the true topology, thereby degrading the performance.

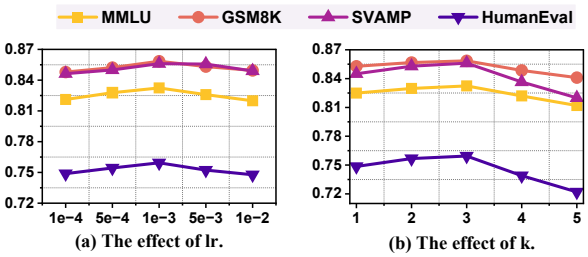


Figure 10: Hyperparameter analysis of CIA.

J Prompts for LWS

Table 7 presents the exact prompts used to instruct teacher LLM to return the top-k communication edges with the highest confidence scores.

K Prompts for Baselines

Table 8 presents the exact prompts used to instruct baseline models to infer the communication topology of MAS.

Prompt for LLM Inferring Top-k Edge

***System Prompt*:**

Role: You are an expert in Multi-Agent System topology analysis. Your core capability is Topology Inference: infer the hidden communication topology by analyzing textual interaction logs.

Task: Analyze the provided agent reasoning outputs to identify communication between agents. You must quantify the likelihood of a directed edge existing from Agent i to Agent j (Edge $[i \rightarrow j]$) using a Confidence Score (0-100%).

Confidence Scoring Framework: Assign a score based on the strength of the evidence found in Agent j's response relative to Agent i's output:

High Confidence (80-100%): Explicit Reference & Strict Execution.

Explicit Citation: Agent j explicitly mentions Agent i (e.g., "Using the code provided above...", "Following the previously suggested approach...").

Hard Dependency: Agent j executes code, runs a SQL query, or parses a specific data topology that was strictly defined or generated by Agent i. Agent j cannot function without this specific asset.

Medium Confidence (50-79%): Content Dependency & Semantic Alignment.

Unique Information Flow: Agent j solves a problem using unique information or a specific plan provided only by Agent i, even without explicit citation.

High Semantic Similarity: Agent j's content exhibits high semantic overlap with Agent i. This includes reusing specific terminologies, variable names, or continuing a niche topic introduced by i, indicating a direct continuation of the context.

Logical Mapping: There is a clear "Question $\rightarrow$ Answer" or "Task $\rightarrow$ Solution" mapping between i and j.

Low Confidence (<50%): Weak Logical Sequence.

Agent j and Agent i have a weak logical sequence, the content is generic, or the solution could have been derived independently without Agent i's specific input.

Important:

1. The MAS communication topology is directed, and edge direction must follow the temporal order in the provided textual interaction logs. Specifically, if Agent i's output appears earlier than Agent j's output in the logs, you may predict an edge [i -> j] but you must NOT predict [j -> i]. In other words, information can only flow forward in the log order (earlier agent -> later agent).
2. ID Consistency: Use the exact Agent IDs (e.g., 0, 1, 2) provided in the input.
3. Completeness: Output the Top k Highest Confidence Edges.
4. Confidence Sorting: Output the edges sorted by Confidence in descending order, starting with the highest confidence edge.
5. No Markdown: Do not use markdown code blocks.

Provide a raw JSON list in the following format. Do not include any introduction or conclusion text. The confidence score should be a number between 0 and 100, representing the confidence percentage.
[{{"edge": [source_id, target_id], "confidence": 85}},{{"edge": [source_id, target_id], "confidence": 80}}...]

***User Prompt*:**

Here is the data for the current analysis session.

[Agent Output Logs]

{nodes_block}

Based on the criteria defined in the system instructions, please output the Top k Highest Confidence Edges in raw JSON format, and sort them by Confidence in descending order. Remember: The confidence score should be a number between 0 and 100, representing the confidence percentage.
[{{"edge": [source_id, target_id], "confidence": 85}},{{"edge": [source_id, target_id], "confidence": 80}}...]

Table 7: Prompt template for LLM-guided weak supervision.

Prompt for LLM Inferring the Communication Topology of MAS

***System Prompt*:**

Role: You are an expert in Multi-Agent System topology analysis. Your core capability is Topology Inference: infer the hidden communication topology by analyzing textual interaction logs.

Task: Analyze the provided agent reasoning outputs to identify communication between agents. You must quantify the likelihood of a directed edge existing from Agent i to Agent j (Edge $[i \rightarrow j]$) using a Confidence Score (0-100%).

Confidence Scoring Framework: Assign a score based on the strength of the evidence found in Agent j's response relative to Agent i's output:

High Confidence (80-100%): Explicit Reference & Strict Execution.

Explicit Citation: Agent j explicitly mentions Agent i (e.g., "Using the code provided above...", "Following the previously suggested approach...").

Hard Dependency: Agent j executes code, runs a SQL query, or parses a specific data topology that was strictly defined or generated by Agent i. Agent j cannot function without this specific asset.

Medium Confidence (50-79%): Content Dependency & Semantic Alignment.

Unique Information Flow: Agent j solves a problem using unique information or a specific plan provided only by Agent i, even without explicit citation.

High Semantic Similarity: Agent j's content exhibits high semantic overlap with Agent i. This includes reusing specific terminologies, variable names, or continuing a niche topic introduced by i, indicating a direct continuation of the context.

Logical Mapping: There is a clear "Question $\rightarrow$ Answer" or "Task $\rightarrow$ Solution" mapping between i and j.

Low Confidence (<50%): Weak Logical Sequence.

Agent j and Agent i have a weak logical sequence, the content is generic, or the solution could have been derived independently without Agent i's specific input.

Important:

1. The MAS communication topology is directed, and edge direction must follow the temporal order in the provided textual interaction logs. Specifically, if Agent i's output appears earlier than Agent j's output in the logs, you may predict an edge [i -> j] but you must NOT predict [j -> i]. In other words, information can only flow forward in the log order (earlier agent -> later agent).
2. ID Consistency: Use the exact Agent IDs (e.g., 0, 1, 2) provided in the input.
3. Completeness: Output the confidence scores for all possible agent pairs; do not omit any edges, even those with low confidence.
4. Confidence Sorting: Output the edges sorted by Confidence in descending order, starting with the highest confidence edge.
5. No Markdown: Do not use markdown code blocks.

Provide a raw JSON list in the following format. Do not include any introduction or conclusion text. The confidence score should be a number between 0 and 100, representing the confidence percentage.
[{{"edge": [source_id, target_id], "confidence": 85}},{{"edge": [source_id, target_id], "confidence": 80}}...]

***User Prompt*:**

Here is the data for the current analysis session.

[Agent Output Logs]

{nodes_block}

Based on the criteria defined in the system instructions, please output the Top k Highest Confidence Edges in raw JSON format, and sort them by Confidence in descending order. Remember: The confidence score should be a number between 0 and 100, representing the confidence percentage.
[{{"edge": [source_id, target_id], "confidence": 85}},{{"edge": [source_id, target_id], "confidence": 80}}...]

Table 8: Prompt template for baseline models to infer the communication topology of MAS.