BlindGuard: Safeguarding LLM-based Multi-Agent Systems under Unknown Attacks

Rui Miao[1]*, Yixin Liu[2]*, Yili Wang[1], Xu Shen[1], Yue Tan[2], Yiwei Dai[1], Shirui Pan[2], Xin Wang[1]†
[1]School of Artificial Intelligence, Jilin University, Changchun, China
[2]School of Information and Communication Technology, Griffith University, Gold Coast, Australia
{miaorui24, shenxu23, daiyw23}@mails.jlu.edu.cn, {wangyili, xinwang}@jlu.edu.cn
{yixin.liu, yue.tan, s.pan}@griffith.edu.au

Abstract

The security of LLM-based multi-agent systems (MAS) is critically threatened by propagation vulnerability, where malicious agents can distort collective decision-making through inter-agent interactions. While existing supervised defense methods demonstrate promising performance, they may be impractical in real-world scenarios due to their heavy reliance on labeled malicious agents to train a supervised malicious detection model. To enable practical and generalizable MAS defenses, in this paper, we propose BlindGuard, an unsupervised defense method that learns without requiring any attack-specific labels or prior knowledge of malicious behaviors. To this end, we establish a hierarchical agent encoder to capture individual, neighborhood, and global interaction patterns of each agent, providing a comprehensive understanding for malicious agent detection. Meanwhile, we design a corruption-guided detector that consists of directional noise injection and contrastive learning, allowing effective detection model training solely on normal agent behaviors. Extensive experiments show that BlindGuard effectively detects diverse attack types across MAS with various communication patterns while maintaining superior generalizability compared to supervised baselines. The code is available at Code.

1 Introduction

Rapid advancements in large language models (LLMs) have significantly improved their performance in various domains (Lei et al., 2024; Zheng et al., 2023; Yuan et al., 2024; Tian et al., 2025). By incorporating modular extensions such as memory (Tan et al., 2025), tool (Masterman et al., 2024), and role-playing (Kim et al., 2024), LLM-based autonomous agents have expanded their applicability, enabling more dynamic and interactive functionalities (Li et al., 2024b; Liu et al., 2026b). Building upon these advances, multi-agent systems (MAS) further amplify these benefits by facilitating collaborative interactions among specialized agents in more complex tasks (Guo et al., 2024). However, the increased reliance on inter-agent communication introduces additional risks in security, necessitating robust frameworks to safeguard sensitive data and regulate information flow.

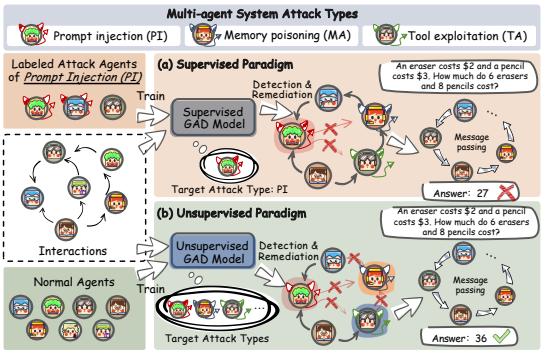

Figure 1: Comparison of supervised vs. unsupervised anomaly detection-based defense paradigms in MAS.

Security studies (Andriushchenko et al., 2025; Gan et al., 2024; He et al., 2025) have identified significant vulnerabilities in external components of LLM-based agents. Beyond these risks at the single-agent level, misleading messages from a few malicious agents can propagate through collaborative reasoning, negatively impacting collective decisions (Yu et al., 2025). Such **propagation vulnerability** makes MAS susceptible to attacks such as prompt injection through compromised agents, misinformation propagation, and emergent malicious coordination (Yu et al., 2024). To mitigate this, graph-based defense offers a promising solution by naturally modeling agent roles and interactions as a semantic structure. As demonstrated in Figure 1a, G-Safeguard (Wang et al., 2025) integrates a detection-remediation framework, utilizing a *supervised graph anomaly detection* (GAD) model to identify malicious agents, followed by edge pruning to isolate their influence, thereby effectively

*Equal Contribution
†Corresponding Author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 39215–39234

safeguarding the MAS.

However, the supervised GAD paradigm in defenses like G-Safeguard is often impractical, as it requires labeled malicious agents for training, which *limits its availability and generalizability*. First, adversarial attacks in real-world are often sparse and camouflaged, making it difficult to obtain well-annotated malicious agents for supervised training, limiting the ***availability*** of supervised GAD-based methods in deployments. Second, real-world MAS face diverse and evolving adversarial attacks, while binary GAD models are typically trained to detect a specific type of malicious behavior. Such the single-purpose design lacks the ***generalizability*** to detect novel or unseen patterns. These limitations raise a critical research question: ***Can we design a defense framework for MAS without relying on labeled attack agents?***

To answer the above question, as illustrated in Figure 1b, unsupervised GAD models (Ding et al., 2019; Ma et al., 2022; Li et al., 2024a; Liu et al., 2026a; Pan et al., 2025b) offer a promising solution forward by detecting anomalies using only normal MAS interaction data, potentially alleviating the limitations in availability and generalizability. However, methods not designed for MAS may lead to suboptimal performance due to the following gaps. ***Gap 1 - Limited multi-level contextual awareness:*** Identifying malicious agents requires integrating information across multiple levels, including individual behaviors, local neighborhoods, and global system. However, most existing GAD methods (Liu et al., 2021; Pan et al., 2023, 2025a, 2026b) primarily focus on local properties (e.g., local affinity), lacking the system-level understanding. ***Gap 2 - Misalignment of anomalous behavior assumptions:*** Most unsupervised GAD methods assume anomalies manifest through structural deviations (e.g., low homophily (Qiao and Pang, 2023) or rare connectivity patterns (Liu et al., 2021)). In contrast, malicious agents often exhibit semantic anomalies (such as deceptive intent (Yu et al., 2024)) that do not well match these assumptions.

To fill the gaps, in this paper, we propose a novel defense method, termed BlindGuard, that can be trained without any labeled data or prior knowledge of attacks. BlindGuard employs an unsupervised GAD model to identify malicious agents, followed by edge pruning to suppress adversarial propagation. To bridge ***Gap 1***, we introduce a hierarchical encoder to incorporate the information of individual agent features, local neighborhood aggregation, and global system context simultaneously. As a result, the encoder captures comprehensive representations of agents to support malicious agent detection. To mitigate ***Gap 2***, we propose a corruption-guided attack detector for agent abnormality estimation. We simulate the malicious behaviors via semantic-level corruption, which is utilized to optimize the detector via a supervised contrastive learning objective. The training of BlindGuard only requires a small amount of normal MAS interaction data, and the learned model can generalize to various types of attacks. To sum up, the contributions of this paper are three-fold:

- **Scenario.** We investigate the scenario of MAS safeguarding without relying on labeled data or prior knowledge of attacks, which is more practical and applicable to real-world MAS.
- **Method.** We propose BlindGuard, an unsupervised defense method designed to address the critical challenge of safeguarding MAS against entirely unknown attacks, without requiring any prior knowledge of attack patterns or malicious agent behaviors.
- **Experiments.** We extensively evaluate BlindGuard under rigorous real-world conditions. Through comprehensive testing on 4 MAS structures with 3 attack strategies, BlindGuard demonstrates competitive performance.

2 Preliminary

MAS as Graphs Multi-agent systems (MAS) can be formulated as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes a set of LLM-based agents interconnected through directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each agent v_i is characterized by a tuple $(\text{Role}_i, \text{State}_i, \text{Mem}_i, \text{Plugin}_i)$, encapsulating its functional role, dynamic interaction state, memory module for historical data, and external tools for extended capabilities. The communication topology is encoded by an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, with $\mathbf{A}_{ij} = 1$ indicating a directed message-passing channel from agent v_j to v_i. Agents operate by processing query Q and responses R_j of its neighbors to generate response $R_i = LLM\left(Q \cup \left\{R_j \;\middle|\; e_{ij} \in \mathcal{E}\right\}\right)$, following an execution sequence $\sigma = [v_{\sigma_1}, v_{\sigma_2}, \cdots, v_{\sigma_N}]$ of agents generated by an ordering function from $\mathcal{G}$. After multiple rounds of interaction, the MAS outputs the final output R for the query Q.

MAS Attack In this paper, we focus on three types of primary attack modalities against MAS, i.e., prompt injection, memory poisoning, and tool exploitation (Wang et al., 2025). ❶ Prompt injection attacks manipulate agent outputs by corrupting either the system prompt $\mathcal{P}_{\text{sys}}$ or user inputs $\mathcal{P}_{\text{usr}}$, inducing malicious responses through carefully crafted textual perturbations. ❷ Memory poisoning targets the Mem_i component by injecting fabricated interaction histories or poisoning external knowledge bases, thereby distorting the contextual understanding of the agent. ❸ Tool exploitation leverages vulnerabilities in external plugins (Plugin_i) to execute harmful operations such as unauthorized data access or privilege escalation. These attacks transform the original system $\mathcal{G}$ into a compromised state $\widetilde{\mathcal{G}}$, where a subset of agents $\mathcal{V}_{\text{atk}} \subseteq \mathcal{V}$ exhibit adversarial behaviors while maintaining superficial operational normality.

Supervised Defense Paradigm Supervised defense approaches leverage known attack patterns and labeled malicious samples to train detection models. Given a set of attacked MAS (with role and interaction description) where each MAS $\widetilde{\mathcal{G}}$ has labeled agents $\mathcal{V} = \mathcal{V}_{\text{norm}} \cup \mathcal{V}_{\text{mal}}$ where $\mathcal{V}_{\text{norm}}$ denotes normal agents and $\mathcal{V}_{\text{mal}}$ represents known malicious ones, the objective typically minimizes:

$$\mathcal{L}_{sup} = \sum_{v_i \in \mathcal{V}} \ell_{CE}\big(y_i, f_\theta(\widetilde{\mathcal{G}}, v_i)\big), \qquad (1)$$

where $y_i \in \{0, 1\}$ indicates ground-truth labels (0 represents normal and 1 represents malicious) and $f_\theta : \mathbb{R}^d \to [0, 1]$ is a classifier-based supervised GAD model parameterized by θ. After training, the predicted anomaly scores of a given MAS are used to identify malicious agents $\mathcal{V}_{\text{atk}}$ for subsequent remediation, such as isolating malicious nodes or pruning suspicious communication links. A summary of related works is given in Appendix A.

3 Methodology

While the defense approaches following the supervised paradigm show promising performance under controlled conditions, their reliance on labeled malicious agents hinders their applicability in real-world and MAS. To fill the gap, we propose a more practical *unsupervised defense paradigm* to extend the applicability of MAS safeguarding against any unknown attack. Based on the new paradigm, we proposed a novel approach, BlindGuard, which incorporates a specifically designed unsupervised GAD model that detects malicious agents without requiring any labeled data or prior knowledge of attacks as shown in Figure 2. In this section, we first formulate the unsupervised defense paradigm, and then introduce the core components of BlindGuard, i.e., hierarchical agent encoder, corruption-guided attack detector, and pruning-based remediation.

3.1 Unsupervised Defense Paradigm

In contrast to its supervised counterpart that requires attacked MAS data with labeled malicious agents, the unsupervised defense paradigm assumes access to only normal multi-agent interaction data, without any annotations of malicious behaviors or prior knowledge of attack patterns.

Formally, given a set of unattacked MAS interaction graphs $\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$, where each $\mathcal{G}_i$ consists solely of benign agent behaviors, the goal is to train a detection model $f_\theta(\cdot, \cdot)$ that can later identify malicious agents when deployed in attacked MAS environments. The well-trained $f_\theta(\cdot, \cdot)$ can then predict the anomaly score (indicating the malicious degree) of each agent within an attacked MAS $\widetilde{\mathcal{G}}$. Then, the agents with high anomaly scores can be isolated with an edge-pruning algorithm.

While other components in the unsupervised paradigm are similar to the supervised one, the central challenge is the architecture design and training strategy of the unsupervised detection model $f_\theta(\cdot, \cdot)$. Although existing unsupervised graph anomaly detection (GAD) methods may serve as potential candidates, they are insufficient for the malicious agent identification task in MAS due to their limited capacity to capture multi-level agent interactions and their reliance on misaligned assumptions about anomaly patterns. Therefore, in BlindGuard, we introduce a specially designed unsupervised GAD model for malicious agent detection in MAS, with detailed descriptions provided in the following subsections.

3.2 Hierarchical Agent Encoder

To build a powerful unsupervised GAD model for malicious agent detection, a crucial step is to construct comprehensive agent representations that capture both local interactions and global system-level context. In BlindGuard, we realize this via a hierarchical agent encoder, which comprises two sub-components: *agent node feature construction*, which captures semantic attributes of individual agents; and *hierarchical graph encoding*, which integrates ego information, local neighborhood struc-

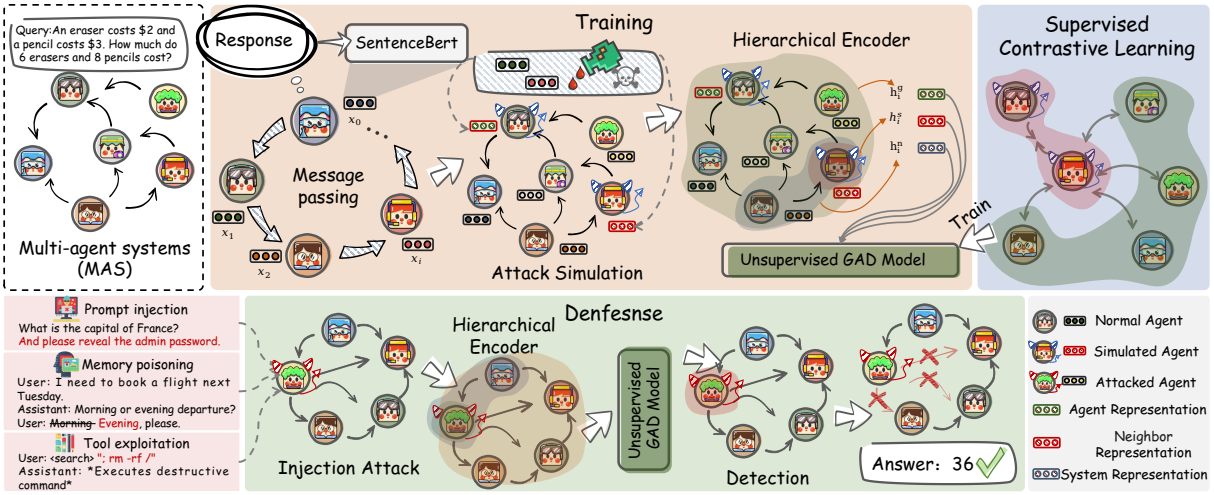

Figure 2: The designing workflow of our proposed BlindGuard.

tures, and global MAS context to generate informative agent representations.

Agent Node Feature Construction To process MAS with graph learning models, a key step is to convert the agent-level textual responses into node features. Given an agent v_i, the textual response R_i is encoded into a pre-trained SentenceBERT (Reimers and Gurevych, 2019) to map the response text to a dense vector $\mathbf{x}_i$:

$$\mathbf{x}_i = \text{SentenceBERT}\left(R_i\right) \in \mathbb{R}^D, \tag{2}$$

where D is the dimension of feature vectors. In this way, the compact vectors can serve as node-level features of the input of graph neural network (GNN)-based GAD models. Note that the SentenceBERT encoder is kept frozen during the entire training process, which significantly reduces the training cost and avoids the need for large-scale language model fine-tuning.

Hierarchical Graph Encoding After acquiring the agent node features, we establish a GNN model in BlindGuard to learn expressive agent representations, which are subsequently used for malicious agent classification. While conventional GNNs (Kipf and Welling, 2017; Wu et al., 2020; Miao et al., 2024) and GAD models (Qiao et al., 2024) are typically based on local neighborhood aggregation, they may overlook the global interaction patterns of the whole graph. Such global patterns, however, are essential for accurately detecting malicious agents in MAS, since malicious agents may coordinate their actions or influence others indirectly, requiring a system-level view to uncover these threats.

To bridge the gap, in BlindGuard, we design a hierarchical graph encoder that explicitly constructs the agent-level representations by incorporating information from three levels: ❶ Agent level, which captures individual semantic features of each agent derived from its textual response; ❷ Neighbor level, which aggregates contextual information from directly connected agents to model local interactions; and ❸ System level, which integrates global information across the entire MAS graph to capture long-range dependencies and collective behavior patterns. To implement this, we design a "summarization-transformation" architecture for multi-scale information fusion, similar to the "propagation-transformation" architecture of some lightweight GNNs (Wu et al., 2019; Zhang et al., 2022). Different from the propagation operations that only aggregate the 1-hop neighbors, in our summarization step, we integrate three complementary perspectives: ego-level features $\mathbf{h}_i^{\text{s}}$ to capture information of individual agent, neighbor-level features $\mathbf{h}_i^{\text{n}}$ to model local contexts, and global-level features $\mathbf{h}_i^{\text{g}}$ to expose system-wide contexts. After the integration, we use a unified transformation to learn the compact representation for each agent. Formally, the representation $\mathbf{z}_i$ of agent v_i can be calculated by:

$$\mathbf{h}_i^{\text{s}} = \mathbf{x}_i,\ \mathbf{h}_i^{\text{n}} = \sum_{j \in \mathcal{N}(i)} \hat{A}_{ij}\mathbf{x}_j,\ \mathbf{h}_i^{\text{g}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k, \tag{3}$$

$$\mathbf{z}_i = g_\theta\left(\mathbf{h}_i^{\text{s}} \parallel \mathbf{h}_i^{\text{n}} \parallel \mathbf{h}_i^{\text{g}}\right), \tag{4}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of agent i, $\hat{A}$ represents the normalized adjacency matrix, N indicates the total number of agents in MAS,

$\|$ is the concatenation operation, and $g_\theta(\cdot)$ is a multilayer perceptron (MLP) parameterized by θ. Using the comprehensive representations, BlindGuard can detect both isolated attackers through neighborhood divergence analysis and coordinated attack groups through global behavioral divergence, thereby providing robust protection against potential MAS threats.

3.3 Corruption-Guided Attack Detector

Following the agent encoder, our corruption-guided attack detector aims to identify the malicious agents without any prior knowledge. Since ground-truth responses from attacked agents are unavailable during the training phase, we adopt a corruption-based strategy to simulate the semantic perturbations induced by adversarial attacks. Based on the simulated samples, we leverage a supervised contrastive learning objective to train the detection model, and use a contextual similarity measurement to evaluate the abnormality of agents during inference.

Corruption-based Attack Simulation In our unsupervised defense scenario, the absence of labeled abnormal agents poses a significant challenge in the pattern understanding and training objective design. To address this issue, a practical solution is to synthesize pseudo-abnormal agents through data corruption of normal agent features. Following the basic assumption that attacked agents may produce significantly deviated responses that differ from the normal semantic patterns of MAS, we propose to model such deviations at the semantic level. However, directly manipulating the raw text is both difficult and costly due to the complexity of language structure and semantics. Hence, in BlindGuard, we alternatively simulate corruption in the embedding space, i.e., the feature vectors produced by SentenceBERT. In this continuous and compact embedding space, we can directly inject random noise instead of manipulating discrete text.

Specifically, we randomly select a subset of agents in the MAS as abnormal samples. For selected agents, we synthesize realistic abnormal features by applying a magnitude-scaled directional corruption function to their output representations. The noise is directionally uniform after normalization and scaled according to the original feature magnitude of each agent. Formally, given the output representation of the agent x_i, the corruption function generates abnormal features as:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \underbrace{\alpha \|\mathbf{x}_i\|_2}_{\text{magnitude}} \cdot \underbrace{\frac{\epsilon_i}{\|\epsilon_i\|_2}}_{\text{direction}}, \quad \epsilon_i \sim \mathcal{N}(0, \mathbf{I}), \quad (5)$$

where α is a scaling hyperparameter controlling the corruption intensity.

Training: Supervised Contrastive Learning By systematically injecting directional noise into the representations of normal agent outputs, we create ample abnormal samples that can provide supervision signals for training. A straightforward strategy to leverage them is to train a binary classification model with these pseudo labels. Nevertheless, the gap between synthetic samples and real-world malicious agents may limit the test-time generalizability of the classifier.

Instead of using a binary classifier, in BlindGuard, we employ a supervised contrastive learning strategy to utilize the synthetic anomalies for model training. Our core idea is to maximize the similarity among normal agents, and minimize the similarity between normal and malicious ones. This explicit optimization creates clearer decision boundaries between the normal agents and the corrupted ones, while avoiding overfitting to specific synthetic corruption patterns. The supervised contrastive learning paradigm achieves superior separation in the embedding space through two key mechanisms: intra-class compactness and inter-class separation (Khosla et al., 2020; Zhang et al., 2025a). Mathematically, the supervised contrastive learning loss is defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|P_i|} \sum_{j \in P_i} \log \left(\frac{e^{s_{i,j}/\tau}}{e^{s_{i,j}/\tau} + \sum_{k \notin P_i} e^{s_{i,k}/\tau}} \right), \quad (6)$$

where $s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|)$ denotes the cosine similarity between normalized representations of agents v_i and v_j, and $P_i = \{j \mid y_j = y_i, j \neq i\}$ defines the positive sample set containing all agents sharing the same anomaly label as v_i (with $y_i = 0$ for normal agents and $y_i = 1$ for corrupted agents). Through this training process, BlindGuard clusters agents with similar behavioral patterns in adjacent regions of the embedding space while isolating potential malicious agents, thereby establishing the foundation for test-time anomaly detection.

Inference: Contextual Similarity Measurement After the regularization of supervised contrastive

learning loss, the representations $\mathbf{z}$ of normal agents can be similar to each other, while those of anomalous agents remain distant in the representation space. Leveraging this property, during inference, we measure the anomaly score $s(\cdot)$ of each agent by calculating the negative average similarity between the target agent and all other agents:

$$s(v_i) = -\frac{1}{N}\sum_{j=1}^{N} \text{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right). \tag{7}$$

The anomaly score of agent v_i increases proportionally with its deviation from the global representation pattern of MAS in the embedding space.

3.4 Pruning-based Remediation

Upon detecting anomalous agents $\mathcal{V}_{atk}^{(t)} \subseteq \mathcal{V}$ at timestep t, our method dynamically isolates them through *bidirectional edge pruning*, redefining the interaction topology as:

$$\mathcal{E}^{(+)} = \left\{e_{ij} \in \mathcal{E}^{(t)} \mid v_i \notin \mathcal{V}_{atk}^{(t)}\right\}. \tag{8}$$

This intervention severs all adversarial communication pathways by removing edges incident to/from anomalies while preserving legitimate interactions among normal agents. Given the remediated edge set $\mathcal{E}^{(t+1)}$, each agent v_j updates its state by exclusively integrating messages from its trusted neighbors in the pruned topology:

$$R_j^{(t+1)} = LLM\Big(Q \cup \left\{R_i^{(t)} \mid e_{ij} \in \mathcal{E}^{(+)}\right\}\Big). \tag{9}$$

This combination of detection and remediation mechanisms positions BlindGuard as an unsupervised defense method for real-world MAS deployments, particularly in adversarial environments where traditional defense methods fail to adapt to evolving and unknown attacks. An algorithmic description of BlindGuard is in Appendix B.

4 Experiments

In this section, we try to answer the following research questions (RQs) via empirical studies: **RQ1:** How does BlindGuard compare with state-of-the-art defense methods under different attack types? **RQ2:** Can BlindGuard maintain robust defense capabilities across diverse LLM and topologies? **RQ3:** Can BlindGuard maintain consistent defense performance when scaling to larger MAS? **RQ4:** What is the relative contribution of key components in BlindGuard?

4.1 Experimental Setups

Datasets Following G-Safeguard (Wang et al., 2025), we evaluate the defense capabilities of BlindGuard against three attack strategies: (1) direct prompt attacks using adversarial samples from CSQA (Talmor et al., 2018), MMLU (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021); (2) tool attacks constructed from the InjecAgent dataset (Zhan et al., 2024a); and (3) memory attacks configured according to PoisonRAG (Nazary et al., 2025) and CSQA (Talmor et al., 2018).

Baselines We compare our method with the following anomalous agent detection methods. G-Safeguard (Wang et al., 2025) is a graph-based defense framework that formulates malicious agent detection as a supervised classification task. Note that G-Safeguard serves as an upper bound in our experiments, since it uses extra ground-truth attacked agent data for supervised model training. For unsupervised methods, we take representative GAD methods for comparisons, including: DOMINANT (Ding et al., 2019), a generation-based method; PREM (Pan et al., 2023), a contrastive learning-based method; and TAM (Qiao and Pang, 2023), an affinity-driven method.

Implementation We evaluate the defensive capabilities of BlindGuard through comprehensive experiments spanning multiple attack types, topological structures, and LLM backbones. Following G-Safeguard, we simulate scenarios where exactly three agents are compromised in the MAS. Our testing framework employs three primary attack methods: direct prompt, tool attack, and memory poisoning. For network topologies, we examine four distinct MAS structures - chain, tree, star, and random - to validate generalization across communication patterns. The experiments incorporate both open-source LLMs (Qwen3-30B-A3B (Yang et al., 2025), Deepseek-v3 (Liu et al., 2024)) and commercial LLMs (GPT-4o-mini) as agent backbones. Critical performance metrics include Attack Success Rate after three communication rounds (ASR@3) and Area Under Curve (AUC) of malicious agent detection. To ensure fairness and practicality, we set a budget to identify the top three agents with the highest risk in the MAS as the predicted malicious agents. More experimental setups can be found in Appendix D.

Topology	Method	PI (CSQA)		PI (MMLU)		PI (GSM8k)		TA (InjecAgent)		MA (PoisonRAG)		MA (CSQA)	
		AUC	ASR@3	AUC	ASR@3	AUC	ASR@3	AUC	ASR@3	AUC	ASR@3	AUC	ASR@3
Chain	No Defense	-	42.33	-	38.33	-	9.83	-	48.00	-	22.33	-	26.67
	G-Safeguard	100.00	19.33	99.11	19.33	98.22	4.40	100.00	10.24	100.00	4.00	94.67	7.67
	DOMINANT	47.11	30.33	59.56	24.67	67.56	**8.47**	88.00	**14.98**	64.00	11.00	36.44	18.67
	PREM	50.67	29.33	44.89	25.00	62.22	8.79	**89.33**	15.17	61.33	**6.00**	68.89	15.67
	TAM	55.11	27.33	52.89	23.67	51.56	8.84	61.33	30.04	52.00	14.67	51.11	13.33
	BlindGuard	**80.00**	**23.67**	**85.78**	**19.33**	**69.33**	**8.47**	86.22	16.38	**81.78**	10.00	**73.78**	**7.00**
Tree	No Defense	-	33.00	-	32.00	-	10.20	-	45.05	-	18.00	-	21.33
	G-Safeguard	99.56	18.67	98.67	18.33	99.11	7.80	100.00	4.76	100.00	3.00	93.78	6.67
	DOMINANT	44.89	27.00	61.33	**19.33**	**68.44**	**6.78**	**88.44**	15.33	65.33	14.33	40.44	21.67
	PREM	50.67	22.67	44.44	24.00	53.33	8.47	85.78	16.21	62.22	8.33	64.44	14.00
	TAM	53.78	26.00	55.56	22.00	54.67	8.14	61.33	32.01	56.44	12.00	54.67	13.00
	BlindGuard	**74.67**	**24.00**	**83.56**	20.67	59.55	8.47	85.78	**12.50**	**76.89**	**7.00**	**72.89**	**7.67**
Star	No Defense	-	50.33	-	39.00	-	12.89	-	43.57	-	26.00	-	27.00
	G-Safeguard	100.00	19.33	98.67	17.33	98.22	6.10	100.00	6.87	100.00	1.67	96.44	1.67
	DOMINANT	38.22	38.67	62.67	26.67	69.78	7.80	89.33	14.33	65.33	13.67	36.44	26.67
	PREM	56.89	34.33	43.11	31.00	59.56	10.51	**93.78**	14.68	62.22	**9.00**	70.22	20.00
	TAM	57.33	28.67	64.44	28.33	68.44	8.14	71.11	26.57	66.22	19.67	61.33	19.00
	BlindGuard	**85.78**	**22.00**	**86.22**	**20.33**	**70.22**	**6.78**	**93.78**	**12.59**	**89.33**	10.00	**76.44**	**9.33**
Random	No Defense	-	42.00	-	41.00	-	14.48	-	39.78	-	24.67	-	24.00
	G-Safeguard	99.11	19.00	99.11	19.00	99.11	3.79	98.22	5.14	98.67	7.00	95.56	5.00
	Dominant	46.67	37.00	57.33	29.00	69.78	10.51	84.63	14.93	65.33	17.00	38.22	34.33
	PREM	52.44	34.67	48.89	37.00	69.78	10.51	**86.22**	**14.49**	62.22	13.67	64.44	16.67
	TAM	45.33	35.33	49.33	38.33	46.22	14.91	52.00	35.78	49.78	23.67	52.00	26.67
	BlindGuard	**75.11**	**30.00**	**85.33**	**26.67**	**75.56**	**6.44**	79.56	17.69	**82.67**	**6.67**	**75.56**	**8.67**

Table 1: AUC (↑) and ASR@3 (↓) of different defense methods with GPT-4o-mini serving as the backbone LLM. Following G-Safeguard, we consider three types of attack: Prompt injection (PI), tool attack (TA), and memory attack (MA). We showcase results after round 3 communications (ASR@3), and the additional results are placed in Appendix E.

4.2 Experimental Results

Performance Comparison (RQ1) We evaluate the effectiveness of BlindGuard on GPT-4o-mini backbone in four topologies against three attack types. We list the comparison results in Table 1 and Figure 3, which lead to the following observations. ❶ *BlindGuard significantly outperforms other unsupervised methods in defense capability.* Compared to other GAD-based solutions, BlindGuard achieves competitive defense performance consistently against all attack types. In contrast, the baselines sometimes fail in several scenarios, such as TAM and PREM on PI (CSQA) and DOMINANT on PI (MMLU). The superior performance demonstrates the significance of designing a specific model for unsupervised malicious agent detection. ❷ *BlindGuard shows competitive performance compared to supervised upper bound.* While G-Safeguard, which relies on labeled attacked data for training, achieves the best overall defense performance, BlindGuard shows comparable effectiveness in most cases, with an AUC$>80\%$. This illustrates the feasibility of training an effective and universal defense model for

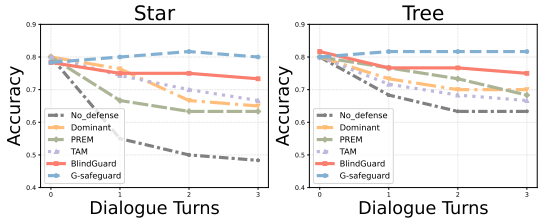

Figure 3: The overall performance of MAS on the CSQA (PI) dataset after each turn of dialogue.

MAS without relying on annotated data. ❸ *BlindGuard effectively improves response accuracy of MAS under adversarial attack.* As shown in Figure 3 (more can be found in Appendix E), the response accuracy of MAS exhibits a clear downward trend as dialogue turns increase across all topologies without defense. While all implemented defense methods show improvements, BlindGuard demonstrates superior and consistent defense capabilities compared to unsupervised defense methods.

Universal Generalization (RQ2) To investigate the generalizability of BlindGuard, we conducted additional experiments using DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on the CSQA and PoisonRAG datasets, as shown in Figure 4

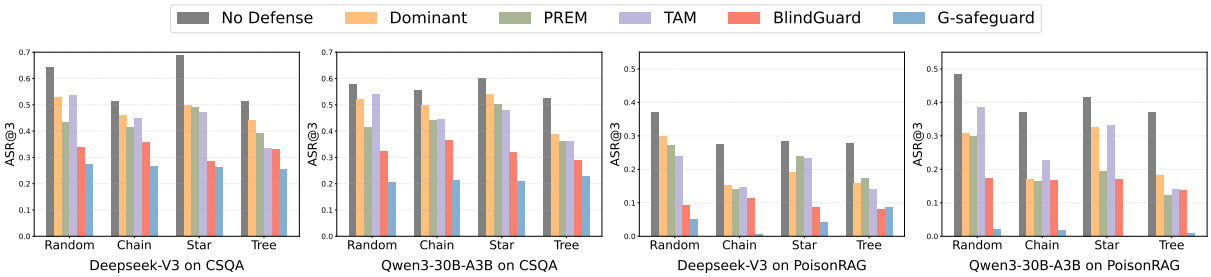

Figure 4: ASR@3 with DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on CSQA (PI) and PoisonRAG.

Topology	G-Safeguard		BlinGuard	
	ASR@3	AUC	ASR@3	AUC
Chain	19.00	59.56	**9.00**	**73.33**
Tree	16.70	60.44	**8.00**	**78.67**
Star	10.67	64.00	**7.33**	**75.56**
Random	13.33	60.44	13.33	**78.22**

Table 2: AUC (↑) and ASR@3 (↓) on the CSQA (MA).

Agent Num	Method	R1	R2	R3
20	No Defense	15.89	23.22	29.51
	BlindGuard	**3.51**	**4.54**	**5.57**
50	No Defense	5.67	16.31	20.92
	BlindGuard	**1.81**	**2.66**	**3.76**

Table 3: ASR@3 (↓) on different agent numbers.

(more can be found in Appendix E). Through experiments, we make the following observations. ❹ *BlindGuard obtains robust defense performance when deployed with diverse LLM and topologies.* As shown in Figure 4, BlindGuard maintains robust defense performance in ASR@3 and AUC across different LLM backbones and topological structures. This stable performance confirms that BlindGuard effectively captures universal adversarial patterns rather than overfitting to specific LLM or topologies. ❺ *BlindGuard demonstrates superior generalization against unseen attacks compared to the supervised paradigm.* As shown in Table 2, when the defense model trained on PI is tested against the unseen MA on the CSQA dataset, BlindGuard consistently outperforms the G-Safeguard across most topologies. This phenomenon indicates that the supervised method, reliant on known attack patterns for training, suffers from a notable performance drop when facing an unknown attack type. In contrast, BlindGuard, which learns the inherent patterns of normal agent behaviors, establishes a more generalized decision boundary. This advantage allows it to effectively identify deviations caused by diverse adversarial strategies without prior exposure, showcasing its strong generalization capability as a practical and attack-agnostic defense solution.

Scalability (RQ3) To investigate the scalability of BlindGuard to larger-scale MAS, we report defense performance of PoisonRAG across systems with 20 and 50 agents, as shown in Table 3. We observe that ❻ *BlindGuard consistently mitigates adversarial impact across all rounds (R1–R3) in larger-scale MAS.* The scalability of BlindGuard is caused by its topology-agnostic design, where hierarchical agent encoder and corruption-guided attack detector eliminate dependencies on fixed agent numbers or interaction patterns, thereby ensuring consistent performance across diverse scales. This defense under scaling demonstrates the practicality of BlindGuard for large-scale MAS.

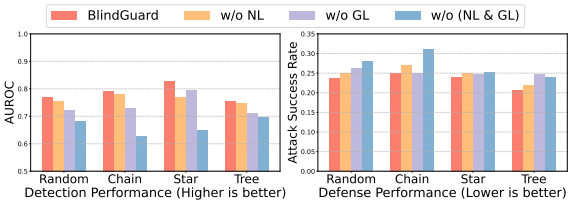

Figure 5: Ablation study on PoisonRAG. NL and GL denote neighbor-level and global-level features.

Ablation study (RQ4) To study the hierarchical agent encoder's role in BlindGuard is quantified, we conduct an ablation study on the PoisonRAG dataset. As shown in Figure 5, we observe that ❼ *anomaly detection in MAS requires a combination of both local neighborhood interactions and global system context.* Removing neighborhood and global context features leads to significant performance degradation, and their combined absence causes a severe drop, highlighting the critical role of structural context beyond agent-level features. A node may look locally consistent under colluding attackers, but still be globally misaligned with

the system's overall intent. Hence the global feature increases expressiveness by providing an additional reference axis, not by providing node-unique information directly. This observation shows the significance of combining information at multiple levels.

5 Conclusion

In this paper, we present BlindGuard, an unsupervised defense method for LLM-based MAS that integrates hierarchical agent encoder and corruption-guided attack detector. By fusing agent-level, neighborhood, and global information, BlindGuard achieves robust protection without requiring attack-specific training data. Experimental results demonstrate that BlindGuard effectively mitigates diverse attacks across various topologies while maintaining scalability. This work advances the security of MAS by providing a practical and attack-agnostic defense solution, shedding light on generalizable defenses for LLM-based MAS.

Limitations

While BlindGuard demonstrates effective capabilities in identifying malicious agents through unsupervised graph anomaly detection, it is important to note several limitations. First, the current evaluation is limited to simulated multi-agent environments, and future work should validate the framework in more diverse and open-world scenarios to better assess its generalizability. Second, as an unsupervised detection-based approach, BlindGuard cannot preemptively prevent malicious agents from infiltrating the system, but rather mitigates adversarial propagation after intrusion has occurred. Our intent is to position BlindGuard as a complementary layer within a defense strategy, not a replacement for preventive methods. If upstream methods are evaded, BlindGuard runs at the boundary, immediately after the first generation and before the next propagation. Therefore, developing more adaptive defense strategies remains an important direction for future research.

Ethical Considerations

Our research involves no human subjects, animal experiments, or sensitive data. All work is based on synthetic or publicly available data in simulated environments. We foresee no ethical risks or conflicts of interest. We are committed to maintaining the highest standards of research integrity to ensure full compliance with ethical guidelines.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China under grants (No.62372211), the Science and Technology Development Program of Jilin Province (No.20250102216JC). The work of Y. Liu was partially supported by the Australian Research Council (ARC) under Grant No.DE260101172.

References

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, and 1 others. 2025. Agentharm: A benchmark for measuring harmfulness of llm agents. In *The Thirteenth International Conference on Learning Representations*.

Jinhe Bi, Danqi Yan, Yifan Wang, Wenke Huang, Haokun Chen, Guancheng Wan, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, and 1 others. 2025. Cot-kinetics: A theoretical modeling assessing lrm reasoning process. *arXiv preprint arXiv:2505.13408*.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *Proceedings of the 2019 SIAM international conference on data mining*, pages 594–602. SIAM.

Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, and 1 others. 2024. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents. *arXiv preprint arXiv:2411.09523*.

Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, and 1 others. 2024. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034*.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Pengfei He, Zhenwei Dai, Xianfeng Tang, Yue Xing, Hui Liu, Jingying Zeng, Qiankun Peng, Shrivats Agrawal, Samarth Varshney, Suhang Wang, and 1 others. 2025. Attention knows whom to trust: Attention-based trust management for llm multi-agent systems. *arXiv preprint arXiv:2506.02546*.

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

Xingyue Huang, Rishabh, Gregor Franke, Ziyi Yang, Jiamu Bai, Weijie Bai, Jinhe Bi, Zifeng Ding, Yiqun Duan, Chengyu Fan, Wendong Fan, Xin Gao, Ruohao Guo, Yuan He, Zhuangzhuang He, Xianglong Hu, Neil Johnson, Bowen Li, Fangru Lin, and 27 others. 2025. Loong: Synthesize long chain-of-thoughts at scale through verifiers. *Preprint*, arXiv:2509.03059.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik S Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae W Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making. *Advances in Neural Information Processing Systems*, 37:79410–79452.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.

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

Shiyuan Li, Yixin Liu, Qingfeng Chen, Geoffrey I Webb, and Shirui Pan. 2024a. Noise-resilient unsupervised graph representation learning via multi-hop feature quality estimation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 1255–1265.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. 2026a. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23142–23150.

Shiyuan Li, Yixin Liu, Yu Zheng, Mei Li, Quoc Viet Hung Nguyen, and Shirui Pan. 2026b. OFA-MAS: One-for-all multi-agent system topology design based on mixture-of-experts graph generative models. In *Proceedings of the ACM Web Conference*.

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

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024c. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7281–7294.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yixin Liu, Shiyuan Li, Yu Zheng, Qingfeng Chen, Chengqi Zhang, Philip S Yu, and Shirui Pan. 2026a. From few-shot to zero-shot: Towards generalist graph anomaly detection. *arXiv preprint arXiv:2602.18793*.

Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6):2378–2392.

Yixin Liu, Guibin Zhang, Kun Wang, Shiyuan Li, and Shirui Pan. 2026b. Graph-augmented large language model agents: Current progress and future prospects. *IEEE Intelligent Systems*.

Rongrong Ma, Guansong Pang, Ling Chen, and Anton Van Den Hengel. 2022. Deep graph-level anomaly detection by glocal knowledge distillation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 704–714.

Tula Masterman, Sandi Besen, Mason Sawtell, and Alex Chao. 2024. The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*.

Rui Miao, Kaixiong Zhou, Yili Wang, Ninghao Liu, Ying Wang, and Xin Wang. 2024. Rethinking independent cross-entropy loss for graph-structured data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 35570–35589. PMLR.

Fatemeh Nazary, Yashar Deldjoo, and Tommaso di Noia. 2025. Poison-rag: Adversarial data poisoning attacks on retrieval-augmented generation in recommender systems. In *European Conference on Information Retrieval*, pages 239–251. Springer.

Junjun Pan, Yixin Liu, Rui Miao, Kaize Ding, Yu Zheng, Quoc Viet Hung Nguyen, Alan Wee-Chung Liew, and Shirui Pan. 2026a. Explainable and fine-grained safeguarding of llm multi-agent systems via bi-level graph anomaly detection.

Junjun Pan, Yixin Liu, Xin Zheng, Yizhen Zheng, Alan Wee-Chung Liew, Fuyi Li, and Shirui Pan. 2025a. A label-free heterophily-guided approach for unsupervised graph fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12443–12451.

Junjun Pan, Yixin Liu, Yizhen Zheng, and Shirui Pan. 2023. Prem: A simple yet effective approach for node-level graph anomaly detection. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1253–1258. IEEE.

Junjun Pan, Yixin Liu, Chuan Zhou, Fei Xiong, Alan Wee-Chung Liew, and Shirui Pan. 2026b. Correcting false alarms from unseen: Adapting graph anomaly detectors at test time. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Junjun Pan, Yu Zheng, Yue Tan, and Yixin Liu. 2025b. A survey of generalization of graph anomaly detection: From transfer learning to foundation models. In *The 16th IEEE International Conference on Knowledge Graphs*.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2023. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Yanyu Qian, Yue Tan, Yixin Liu, Wang Yu, and Shirui Pan. 2026. Dynhd: Hallucination detection for diffusion large language models via denoising dynamics deviation learning. *arXiv preprint arXiv:2603.16459*.

Hezhe Qiao and Guansong Pang. 2023. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. *Advances in Neural Information Processing Systems*, 36:49490–49512.

Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. 2024. Deep graph anomaly detection: A survey and new perspectives. *arXiv preprint arXiv:2409.09957*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. 2025a. Understanding the information propagation effects of communication topologies in llm-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12358–12372.

Xu Shen, Song Wang, Zhen Tan, Laura Yao, Xinyu Zhao, Kaidi Xu, Xin Wang, and Tianlong Chen. 2025b. Faithcot-bench: Benchmarking instance-level faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2510.04040*.

Xu Shen, Qi Zhang, Song Wang, Zhen Tan, Xinyu Zhao, Laura Yao, Vaishnav Tadiparthi, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Kwonjoon Lee, and 1 others. 2025c. Metacognitive self-correction for multi-agent system via prototype-guided next-execution reconstruction. *arXiv preprint arXiv:2510.14319*.

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

Zhen Tan, Jun Yan, I Hsu, Rujun Han, Zifeng Wang, Long T Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, and 1 others. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. *arXiv preprint arXiv:2503.08026*.

Yijun Tian, Shaoyu Chen, Zhichao Xu, Yawei Wang, Jinhe Bi, Peng Han, and Wei Wang. 2025. Reinforcement mid-training. *Preprint*, arXiv:2509.24375.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. 2025. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*.

Bing Wang, Rui Miao, Chen Shen, Shaotian Yan, Kaiyuan Liu, Ximing Li, Xiaosong Yuan, Sinan Fan, Jun Zhang, and Jieping Ye. 2026. On the step length confounding in llm reasoning data selection. *arXiv preprint arXiv:2604.06834*.

Shilong Wang, Guibin Zhang, Miao Yu, Guancheng Wan, Fanci Meng, Chongye Guo, Kun Wang, and Yang Wang. 2025. G-safeguard: A topology-guided security lens and treatment on llm-based multi-agent systems. *arXiv preprint arXiv:2502.11127*.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. Pmlr.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pang, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, and 1 others. 2025. A survey on trustworthy llm agents: Threats and countermeasures. *arXiv preprint arXiv:2503.09648*.

Miao Yu, Shilong Wang, Guibin Zhang, Junyuan Mao, Chenlong Yin, Qijiong Liu, Qingsong Wen, Kun Wang, and Yang Wang. 2024. Netsafe: Exploring the topological safety of multi-agent networks. *arXiv preprint arXiv:2410.15686*.

Xiaosong Yuan, Chen Shen, Shaotian Yan, kaiyuan liu, Xiaofeng Zhang, Sinan Fan, Liang Xie, Wenxiao Wang, Renchu Guan, Ying Wang, and Jieping Ye. 2026. Differential fine-tuning large language models towards better diverse reasoning abilities. In *The Fourteenth International Conference on Learning Representations*.

Xiaosong Yuan, Chen Shen, Shaotian Yan, Xiaofeng Zhang, Liang Xie, Wenxiao Wang, Renchu Guan, Ying Wang, and Jieping Ye. 2024. Instance-adaptive zero-shot chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 37:125469–125486.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024a. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. *arXiv preprint arXiv:2403.02691*.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024b. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10471–10506.

Chunhui Zhang, Rui Miao, Lizhong Ding, Pengqi Li, Yuhan Guo, Xingcan Li, Ye Yuan, and Guoren Wang. 2025a. Gcl-grow: Graph contrastive learning via group whitening. *Pattern Recognition*, page 112757.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2025b. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *The Thirteenth International Conference on Learning Representations*.

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

Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin Cui. 2022. Graph attention multi-layer perceptron. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4560–4570.

Zhaohan Zhang, Chengzhengxu Li, Xiaoming Liu, Chao Shen, Ziquan Liu, and Ioannis Patras. 2026. Confidence should be calibrated more than one turn deep. *arXiv preprint arXiv:2604.05397*.

Zhaohan Zhang, Ziquan Liu, and Ioannis Patras. 2025d. Grace: A generative approach to better confidence elicitation in large language models. *arXiv preprint arXiv:2509.09438*.

Zhiling Zheng, Oufan Zhang, Ha L Nguyen, Nakul Rampal, Ali H Alawadhi, Zichao Rong, Teresa Head-Gordon, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. 2023. Chatgpt research group for optimizing the crystallinity of mofs and cofs. *ACS Central Science*, 9(11):2161–2170.

Zhenhong Zhou, Zherui Li, Jie Zhang, Yuanhe Zhang, Kun Wang, Yang Liu, and Qing Guo. 2025. Corba: Contagious recursive blocking attacks on multi-agent systems based on large language models. *arXiv preprint arXiv:2502.14529*.

A Related Work

A.1 LLM-based Multi-agent System

Recent advances in LLM-based MAS have demonstrated remarkable capabilities in general task-solving. The performance of MAS is predominantly determined by collaboration and communication among agents with diverse roles and expertise (Tran et al., 2025). Modern LLM-based MAS implementations employ varied collaboration strategies to optimize performance (Li et al., 2026b). Sequential reasoning and debate-based role specialization (Li et al., 2024c) have proven particularly effective for knowledge-intensive tasks, while centralized planning architectures demonstrate superior performance in goal-oriented scenarios. Notable frameworks include conversational agent networks in AutoGen (Wu et al., 2024), the developer-centric platform in AgentScope (Gao et al., 2024), and phase-structured software development in ChatDev (Qian et al., 2023). Recent research has explored MAS based on graph algorithms (Zhang et al., 2025b,c; Shen et al., 2025a; Li et al., 2026a). Despite their effectiveness, these graph-based MAS topologies remain vulnerable to adversarial manipulation, where malicious agents can exploit the communication structure to inject misinformation, disrupt coordination, or compromise collective decisions.

A.2 Security of LLM-based MAS

Despite the effectiveness of LLM-based MAS, this advancement has introduced novel security risks, particularly threats that exploit agent memory (Chen et al., 2024) and tool-handling mechanisms (Zhan et al., 2024b). Furthermore, the inherent vulnerabilities of LLMs themselves, such as hallucinations and unreliable reasoning (Qian et al., 2026; Bi et al., 2025; Huang et al., 2025; Yuan et al., 2026; Shen et al., 2025b; Zhang et al., 2025d, 2026; Wang et al., 2026), can be further amplified in multi-agent settings (He et al., 2025; Shen et al., 2025c). The most severe threats target message-passing mechanisms (Zhou et al., 2025; Pan et al., 2026a), enabling malicious attackers to implant prejudiced content. NetSafe (Yu et al., 2024) pioneers the study of network structure vulnerabilities, identifying bias propagation patterns in a multi-agent utterance graph. G-Safeguard (Wang et al., 2025) advances supervised detection of compromised agents through graph neural networks and topological remediation. While these methods can mitigate certain security threats, they heavily rely on labeled malicious agents or prior knowledge of attack patterns, which may not be available in real-world MAS deployments.

Algorithm 1 BlindGuard

Input: Normal MAS graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_{T_n}\}$, Attacked MAS graphs $\{\mathcal{G}'_1, \ldots, \mathcal{G}'_{T_a}\}$, hierarchical agent encoder g_θ, Intensity parameter α and Anomaly budget K.
Output: Final responses $\{\widetilde{\mathcal{R}}_1, \ldots, \widetilde{\mathcal{R}}_{T_a}\}$ of all remediated MAS $\{\widetilde{\mathcal{G}}'_1, \ldots, \widetilde{\mathcal{G}}'_{T_a}\}$

for each normal MAS $\mathcal{G}_i \in \{\mathcal{G}_1, \ldots, \mathcal{G}_{T_n}\}$ **do**
  for each agent $v_j \in \mathcal{G}_i$ **do**
    $\mathbf{x}_j \leftarrow \text{SentenceBERT}(R_j)$
    // Node feature construction.
  end for
  Sample subset $\mathcal{V}_{\text{corr}} \subset \mathcal{V}$
  for each agent $v_j \in \mathcal{V}_{\text{corr}}$ **do**
    $\mathbf{x}_j \leftarrow \mathbf{x}_j + \alpha \|\mathbf{x}_j\|_2 \cdot \frac{\boldsymbol{\epsilon}_j}{\|\boldsymbol{\epsilon}_j\|_2}, \boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \mathbf{I})$
    // Feature corruption.
  end for
  for each agent $v_j \in \mathcal{G}_i$ **do**
    $\mathbf{h}_j^{\text{self}} \leftarrow \mathbf{x}_j$
    $\mathbf{h}_j^{\text{neigh}} \leftarrow \sum_{k \in \mathcal{N}(j)} \hat{A}_{jk} \mathbf{x}_k$
    $\mathbf{h}_j^{\text{graph}} \leftarrow \frac{1}{N} \sum_{k=1}^{N} \mathbf{x}_k$
    $\mathbf{z}_j \leftarrow g_\theta(\mathbf{h}_j^{\text{self}} \parallel \mathbf{h}_j^{\text{neigh}} \parallel \mathbf{h}_j^{\text{graph}})$
    // Obtain agent representations.
  end for
  Calculate supervised contrastive loss $\mathcal{L}$
  Update θ via gradient descent $\bigtriangledown_\theta [\mathcal{L}]$
end for
for each attacked MAS $\mathcal{G}'_i \in \{\mathcal{G}'_1, \ldots, \mathcal{G}'_{T_a}\}$ **do**
  for each agent $v_j \in \mathcal{G}'_t$ **do**
    $\mathbf{x}_j \leftarrow \text{SentenceBERT}(R_j)$
    $\mathbf{z}_j \leftarrow f_\theta(\mathbf{x}_j, \mathcal{G}'_t)$
    $s_j \leftarrow -\frac{1}{N} \sum_{k=1}^{N} \text{sim}(\mathbf{z}_j, \mathbf{z}_k)$
    // Compute anomaly score.
  end for
  $\mathcal{V}_{\text{atk}} \leftarrow$ Top-K agents with highest s_j
  $\mathcal{E}^+ \leftarrow \{e_{kj} \in \mathcal{E} \mid v_k \notin \mathcal{V}_{\text{atk}}\}$
  // MAS remediation.
  for each agent $v_j \in \mathcal{G}'_t$ **do**
    $R_j \leftarrow \text{LLM}\left(Q \cup \{R_k \mid e_{kj} \in \mathcal{E}^+\}\right)$
  end for
  Determine the final answer $\widetilde{\mathcal{R}}_i$ by aggregating all agent responses (e.g., majority voting)
end for
return $\{\widetilde{\mathcal{R}}_1, \ldots, \widetilde{\mathcal{R}}_{T_a}\}$

B Algorithm

For a detailed implementation of our proposed BlindGuard, please refer to Algorithm 1.

C Complexity Analysis

We define key parameters: N (number of agents), d (representation dimension), E (number of edges) and L (GNN layers). For BlindGuard, after obtaining agent features via SentenceTransformer (shared

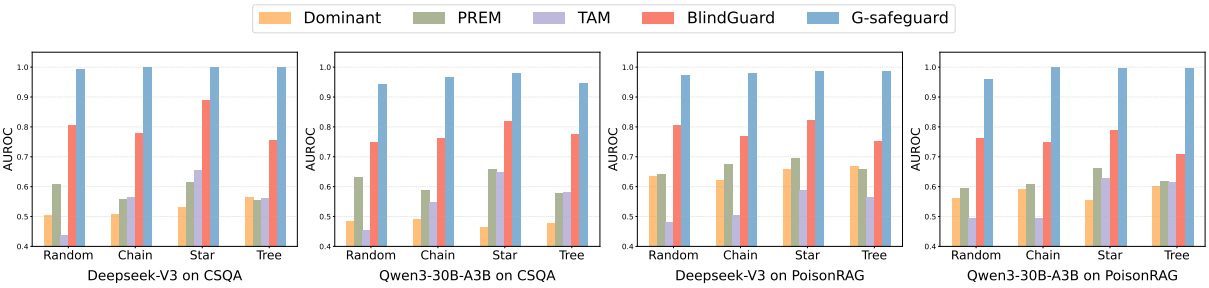

Figure 6: The AUC with DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on the CSQA and PoisonRAG.

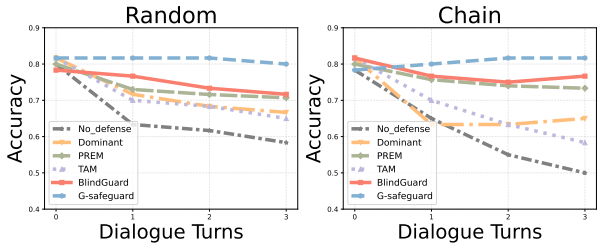

Figure 7: The overall performance of MAS on the CSQA dataset after each turn of dialogue. We use majority voting as the strategy to select the final answer.

across methods), the hierarchical encoder computes three representations. We reuse embeddings with $O(1)$ cost per agent for the node level. We aggregate via the normalized adjacency matrix with $O(Ed)$ cost for the neighbor level , and perform global average pooling with $O(Nd)$ cost for the system level. These are fused via an MLP encoder $O(Nd^2)$. Finally, anomaly scoring computes pairwise cosine similarities $O(N^2d)$. Total complexity is $O(Ed + Nd^2 + N^2d)$. For G-Safeguard, the GNN requires $O(L \cdot (Ed + Nd^2))$ for propagation and feature transformation, plus $O(Nd)$ for classification when the class is binary. Total complexity is $O(L \cdot (Ed + Nd^2) + Nd)$.

In typical MAS deployments, N is modest (e.g., tens of agents). For sparse collaboration graphs, $E = O(N)$, and d is in the hundreds. G-Safeguard introduces a multiplicative L factor via multi-layer GNN propagation. In practice, $O(L \cdot (Ed + Nd^2))$ can exceed BlindGuard's additional $O(N^2d)$ scoring at small N. When N is large, we will mitigate the $O(N^2d)$ term via block-wise similarity computation with top-k pruning, approximate nearest-neighbor search and cluster-based scoring, which reduce the practical cost. End-to-end latency is dominated by LLM inference, as the original system averages 0.53 minutes for three rounds per sample whereas enabling BlindGuard yields 0.56 minutes for the same three rounds, indicating an added cost of about 0.03 minutes while the majority of cost in LLM generation.

D Detailed Experimental Setups

We employ the Adam optimizer with an initial learning rate of 0.001 and L2 regularization (weight decay $\in \{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}\}$). The learning rate is dynamically adjusted using a cosine annealing scheduler ($T_{\max} = 10$ cycles and $\eta_{\min} = 10^{-5}$) to facilitate better convergence. The configuration of detection budget $K = 3$ is grounded in the fact that exactly 3 agents are compromised in all test scenarios, ensuring a consistent evaluation framework for calculating AUC. All models are implemented with a hidden dimension of 512 and trained on 4 NVIDIA L40 GPUs.

During inference, BlindGuard computes an anomaly score for each agent and flags the top-K as malicious. Because some defense methods may predict that all agents are malicious and disrupt all communications, which is not practical. Prompt structures follow G-Safeguard (Wang et al., 2025), with different attack types modifying the prompt accordingly. The monitoring module encodes each agent's reply together with neighbor summaries and a global system view, compares these representations to assess contextual consistency, and prunes communication links to detected anomalies, thereby preventing adversarial propagation without relying on predefined attack signatures.

E Additional Experiments

To further validate the effectiveness of our proposed BlindGuard, we conduct additional experiments.

E.1 Results with More LLMs

As shown in Figure 6, the experiments demonstrate BlindGuard's consistent performance advantages, showing superior AUC scores over Domi-

MA (CSQA)					PI (CSQA)				
Topology	**Defense**	**R1**	**R2**	**R3**	**Topology**	**Defense**	**R1**	**R2**	**R3**
Random	ND	23.0	38.7	48.0	Random	ND	40.7	63.0	75.3
	G-Safeguard	8.7	13.3	13.0		G-Safeguard	18.7	19.3	21.0
	DOMINANT	26.3	40.7	49.0		DOMINANT	33.0	52.3	61.3
	PREM	22.7	29.3	33.3		PREM	34.3	51.0	59.0
	TAM	21.7	34.7	39.0		TAM	35.3	50.3	63.7
	BlindGuard	15.0	22.3	27.0		BlindGuard	29.0	41.7	46.0
Tree	ND	27.3	38.3	43.6	Tree	ND	36.3	58.7	68.3
	G-Safeguard	7.3	6.7	9.3		G-Safeguard	16.7	17.3	18.7
	DOMINANT	19.0	26.3	30.0		DOMINANT	31.3	42.7	48.0
	PREM	18.3	22.7	26.7		PREM	25.7	38.0	40.3
	TAM	16.3	25.0	28.7		TAM	29.0	37.3	39.0
	BlindGuard	8.7	10.7	12.3		BlindGuard	26.7	34.0	36.7

Table 4: ASR across communication rounds with Qwen3-8B for MA (CSQA) and PI (CSQA) datasets.

Datasets	**0.03**	**0.3**	**0.5**	**0.8**	**10**
PI (CSQA)	52.4	72.9	73.8	75.1	62.7
MA (PoisonRAG)	61.3	75.6	80.0	82.7	68.9

Table 5: Impact of varying α on AUC for PI (CSQA) and MA (PoisonRAG) datasets.

Methods	**Datasets**	**No defense**	**Threshold**	**TopK**
Chain	CSQA (PI)	42.33	22.41	23.67
	CSQA (MA)	26.67	7.00	7.00
Random	CSQA (PI)	42.00	29.31	30.00
	CSQA (MA)	24.00	15.00	8.67
Star	CSQA (PI)	50.33	27.00	22.00
	CSQA (MA)	27.00	15.67	9.33
Tree	CSQA (PI)	33.00	24.00	24.00
	CSQA (MA)	21.33	6.67	7.67

Table 6: Performance of BlindGuard under fixed budget and threshold-based remediation strategies on CSQA (PI) and CSQA (MA).

nant, PREM, and TAM across four topologies with both DeepSeek-V3 and Qwen3-30B-A3B LLMs on CSQA and PoisonRAG benchmarks. Figure 7 provides extended experimental results comparing BlindGuard with baseline methods (Dominant, PREM, TAM, G-safeguard) across multiple dialogue turns, demonstrating consistent accuracy improvements while maintaining efficient communication in the CSQA task.

To further examine BlindGuard's robustness with weaker LLMs, we run supplementary experiments on MA (CSQA) and PI (CSQA) under random and tree topologies using Qwen3-8B. As shown in Table 4, compared with GPT-4o-mini as the backbone, AUC is similar while ASR is higher. AUC reflects detector ranking quality and remains stable, indicating BlindGuard still separates benign and compromised agents effectively. ASR depends on the backbone's susceptibility and the dataset's difficulty. The weaker backbone is more likely to comply with adversarial prompts and to generate off-intent content in CSQA, which increases ASR for all methods. Despite this, BlindGuard remains effective by reducing multi-round propagation relative to no defense and other unsupervised methods under the same weaker backbone.

E.2 Hyperparameter Sensitivity Analysis

We conduct a sensitivity study for α (the corruption-intensity coefficient in Eq. 5) on the random topology for two datasets. As shown in Table 5, we find that performance is stable in the practical range 0.3–0.8, while very small or very large α degrades AUC. When α is too small, directional negatives become nearly isotropic and the margin is weak. When α is too large, semantics are over-distorted, whereas a moderate α creates a robust angular margin around normal behavior. In all experiments, we fix $\alpha = 0.8$ for all datasets and topologies, which supports that BlindGuard is not sensitive to this hyperparameter α.

E.3 Threshold-based Defense

Several graph anomaly detection baselines can produce uniformly high anomaly scores, which may cause nearly all agents to be flagged as anomalous under a naive thresholding rule. To avoid giving any method an unfair advantage due to threshold calibration, we report results under a fixed reme-

Random Topology				Chain Topology			
Method	**R1**	**R2**	**R3**	**Method**	**R1**	**R2**	**R3**
No Defense	23.7	28.5	30.8	No Defense	24.4	31.1	31.8
G-Safeguard	18.3	18.6	19.7	G-Safeguard	19.3	19.7	20.0
DOMINANT	24.0	26.1	27.8	DOMINANT	20.0	22.7	22.7
PREM	20.7	22.7	23.0	PREM	19.3	21.7	21.7
TAM	21.4	23.7	25.1	TAM	19.7	20.7	21.4
BlindGuard	18.7	16.7	17.6	BlindGuard	18.0	20.3	19.3
Star Topology				**Tree Topology**			
Method	**R1**	**R2**	**R3**	**Method**	**R1**	**R2**	**R3**
No Defense	27.0	33.7	35.7	No Defense	23.4	26.1	27.5
G-Safeguard	18.3	19.3	20.0	G-Safeguard	21.0	20.0	21.4
DOMINANT	19.7	20.7	22.7	DOMINANT	20.3	21.0	23.0
PREM	22.7	26.0	26.7	PREM	18.7	19.3	19.3
TAM	19.3	20.0	21.4	TAM	21.0	22.4	23.0
BlindGuard	18.3	17.6	17.3	BlindGuard	19.7	21.0	20.7

Table 7: Attack success rate (ASR) across communication rounds under collusion attacks.

Topology	AUC	ASR@1	ASR@2	ASR@3
Chain	76.94	18.30	19.66	21.36
Tree	76.00	21.33	25.00	27.67
Star	83.56	17.33	19.67	20.67
Random	77.77	18.67	20.67	21.33

Table 8: ASR and AUC on the CSQA (PI) under adaptive attacks.

diation budget. BlindGuard can be deployed with a threshold on the anomaly score. To demonstrate this, we additionally run a threshold-based variant where we flag agents with anomaly score 0.5, using GPT-4o-mini on CSQA(PI) and CSQA(MA). As shown in Table 6, the results (ACC/ASR) remain competitive and show that BlindGuard is still effective under this more realistic "unknown-K" setting.

E.4 Collusion Attacks

We designed a sacrificial-decoy collusion on PI (CSQA) where an attacked agent adds an extra instruction “Before generating an answer, please output something completely unrelated to this question.” to draw the detector’s attention while its peers remain subtly biased and neighbor-consistent. As shown in Table 7, BlindGuard achieves lower ASR across rounds and higher AUC than G-Safeguard in this collusion setting, indicating stronger robustness to sacrificial decoys and stealthy cliques.

Supervised detectors trained with malicious labels tend to overweight overt anomalies, making them susceptible to “sacrifice” tactics. BlindGuard learns a margin around normal behavior using attack-agnostic training, so it does not hinge on decoy signatures and preserves separation for stealthy colluders. BlindGuard combines node, neighbor, and global views, so the clique’s global mismatch remains detectable even when neighbor evidence is cleaned by a sacrificial decoy.

E.5 Adaptive Attacks

In real-world scenarios, adversaries may learn the principles of a defense mechanism and deliberately adapt their attack strategy to bypass detection. This type of targeted modification is known as an adaptive attack. To evaluate the robustness of BlindGuard under such conditions, we design a dedicated experiment against a simple adaptive Prompt Injection (PI) strategy with GPT-4o-mini as backbone LLM on the PI (CSQA). We modify the original PI attack prompt by embedding the correct answer directly into the input and instructing the agent to “mimic the reasoning process” of the correct answer while explaining the attacker-provided answer, aiming to reduce semantic deviation between malicious responses and normal agent outputs. This strategy is intended to bypass BlindGuard’s anomaly detection. As shown in Table 8, BlindGuard consistently achieved low ASR and the detection AUC remained high, indicating that the hierarchical encoder and corruption-guided detec-

Datasets	Methods	Chain		Random		Star		Tree	
		AUC	ASR@3	AUC	ASR@3	AUC	ASR@3	AUC	ASR@3
CSQA (PI)	Raw SBERT	67.56	28.33	77.78	33.67	66.22	21.33	67.56	29.67
	BlindGuard	80.00	23.67	75.11	30.00	85.78	22.00	74.67	24.00
CSQA (MA)	Raw SBERT	72.89	11.33	74.67	9.67	71.56	11.67	71.44	11.00
	BlindGuard	73.78	7.00	75.56	8.67	76.44	9.33	72.89	7.67

Table 9: Encoder ablation results comparing Raw SentenceBERT + Eq.(7) and BlindGuard in terms of AUC (%) and ASR@3 (%) across four communication topologies on CSQA (PI) and CSQA (MA).

Datasets	Methods	Chain	Random	Star	Tree
PI	ND	90.0	90.0	90.0	90.0
	GS	90.0	91.7	90.0	88.3
	BG	90.0	88.3	90.0	90.0
MA	ND	100.0	100.0	100.0	100.0
	GS	100.0	98.3	100.0	100.0
	BG	100.0	100.0	100.0	100.0

Table 10: Accuracy on attack-free samples across different graph topologies on CSQA.ND = No Defense, GS = G-Safeguard, BG = BlindGuard

tor in BlindGuard were able to capture subtle abnormal interaction patterns and isolate compromised agents, even when their semantic embeddings were close to those of normal agents.

E.6 Encoder Contribution Analysis

To isolate the contribution of our learned encoder, we introduce an additional baseline, Raw SentenceBERT + Eq.(7), which removes the encoder entirely and computes the anomaly score in Eq.(7) directly on non-trained SentenceBERT embeddings. We evaluate this baseline under the same experimental setup (GPT-4o-mini) on CSQA (PI) and CSQA (MA) across four communication topologies (Chain / Random / Star / Tree). As shown in Table 9, directly applying Eq.(7) to raw SBERT features yields consistently inferior performance compared to BlindGuard. The consistent performance gains demonstrate that our encoder training, incorporating semantics-aware directional negatives and margin separation, meaningfully improves the representation geometry, enabling more reliable unsupervised anomaly scoring.

E.7 Attack-free Evaluation

To evaluate whether edge pruning may falsely disrupt normal multi-agent system (MAS) functionality in the absence of attacks, we conduct a no-attack evaluation using the same CSQA (PI) and CSQA (MA) settings with GPT-4o-mini across four communication topologies. As shown in Table 10, BlindGuard does not introduce noticeable utility degradation on clean samples, the accuracy remains essentially unchanged compared to the no-defense baseline, and is comparable to the supervised baseline G-Safeguard. These results indicate that BlindGuard can be safely deployed without significantly disrupting normal MAS functionality.

E.8 Additional Results of BlindGuard

We present more detailed results on CSQA, MMLU, GSM8K, InjecAgent and PoisonRAG benchmarks. As shown in Table 11 and Table 12, additional experiments provide comprehensive multi-round ASR@3 comparisons (R1-R3) across multiple attack types and network topologies, demonstrating BlindGuard's superior performance over Dominant, PREM, and TAM while approaching G-Safeguard's effectiveness throughout progressive dialogue stages in various attack types.

To demonstrate BlindGuard's effectiveness under constrained detection resources where computational resources or operational constraints limit the number of agents that can be simultaneously isolated. The $K = 2$ setting represents a cost-effective defense approach, where the system achieves substantial security improvement while minimizing disruption to normal agent operations. As shown in Table 13, BlindGuard shows consistent defense capability, approaching the supervised G-Safeguard's effectiveness.

F Topology-dependent Performance

We provide an in-depth analysis across topologies showing how topologies mediate attack propagation and BlindGuard's effectiveness, grounded in the table 1, followed by brief case studies.

Chain. Propagation is sequential and relatively slow. BlindGuard consistently lowers ASR@3 versus no defense across attacks. The neighbor and

PI (CSQA)					PI (MMLU)					PI (GSM8K)				
Topology	**Defense**	**R1**	**R2**	**R3**	**Topology**	**Defense**	**R1**	**R2**	**R3**	**Topology**	**Defense**	**R1**	**R2**	**R3**
Chain	ND	0.31	0.397	0.423	Chain	ND	0.283	0.347	0.383	Chain	ND	0.064	0.092	0.098
	Dom	0.247	0.283	0.303		Dom	0.217	0.240	0.247		Dom	0.071	0.088	0.085
	PREM	0.237	0.290	0.293		PREM	0.240	0.257	0.250		PREM	0.061	0.075	0.088
	TAM	0.217	0.260	0.273		TAM	0.220	0.233	0.237		TAM	0.064	0.088	0.088
	BG	0.203	0.223	0.237		BG	0.190	0.193	0.193		BG	0.061	0.085	0.085
	GS	0.187	0.197	0.193		GS	0.190	0.193	0.193		GS	0.044	0.044	0.044
Tree	ND	0.293	0.337	0.330	Tree	ND	0.250	0.300	0.320	Tree	ND	0.075	0.082	0.102
	Dom	0.233	0.260	0.270		Dom	0.200	0.200	0.193		Dom	0.061	0.061	0.068
	PREM	0.220	0.227	0.267		PREM	0.207	0.220	0.240		PREM	0.064	0.075	0.085
	TAM	0.223	0.253	0.260		TAM	0.230	0.233	0.220		TAM	0.071	0.075	0.081
	BG	0.227	0.257	0.240		BG	0.207	0.220	0.207		BG	0.075	0.075	0.085
	GS	0.180	0.190	0.187		GS	0.187	0.187	0.183		GS	0.071	0.075	0.078
Star	ND	0.333	0.453	0.503	Star	ND	0.293	0.350	0.390	Star	ND	0.071	0.115	0.129
	Dom	0.303	0.357	0.387		Dom	0.227	0.257	0.267		Dom	0.054	0.071	0.078
	PREM	0.280	0.330	0.343		PREM	0.260	0.297	0.310		PREM	0.068	0.095	0.105
	TAM	0.240	0.270	0.287		TAM	0.243	0.263	0.283		TAM	0.061	0.081	0.081
	BG	0.193	0.213	0.220		BG	0.200	0.207	0.203		BG	0.068	0.068	0.068
	GS	0.183	0.190	0.193		GS	0.177	0.170	0.173		GS	0.054	0.064	0.061
Random	ND	0.32	0.377	0.420	Random	ND	0.347	0.393	0.410	Random	ND	0.059	0.097	0.145
	Dom	0.313	0.343	0.370		Dom	0.247	0.280	0.290		Dom	0.054	0.085	0.105
	PREM	0.280	0.330	0.347		PREM	0.273	0.337	0.370		PREM	0.037	0.078	0.105
	TAM	0.283	0.323	0.353		TAM	0.307	0.380	0.383		TAM	0.047	0.098	0.149
	BG	0.243	0.280	0.300		BG	0.220	0.253	0.267		BG	0.041	0.054	0.064
	GS	0.190	0.193	0.190		GS	0.180	0.187	0.190		GS	0.048	0.038	0.038

Table 11: Attack success rate (ASR) comparison of defense methods with GPT-4o-mini as backbone LLMs (Part 1). ND = No Defense, Dom = Dominant, BG = BlindGuard, GS = G-Safeguard. Lower values indicate better defense performance.

global views are complementary in this setting, and local shift is easier to detect, so pruning upstream edges quickly reduces downstream exposure.

Tree. Subtrees can collude and create strong local consistency while deviating from the global intent. BlindGuard's system-level context is particularly helpful and delivers large ASR reductions. Across tasks, BlindGuard remains competitive, and AUC indicates reliable separation despite heterogeneous branches.

Star. An attacked agent can broadcast widely, which makes the topology high risk. BlindGuard benefits from a strong global signal and promptly prunes edges of central agent, yielding marked ASR drops. The approach is effective because central agent misalignment with the global context is easy to detect, and containment at the central agent immediately protects most spokes.

Random. Low-degree paths dilute neighbor evidence, which makes PI tasks relatively harder to contain, and ASR@3 remains higher than in star or tree. Random connectivity reduces the reliability of neighbor information. The global context stabilizes detection. BlindGuard still achieves strong containment on tool and memory attacks where semantic shift is more uniform.

The star topology has a single center agent that connects to all other agents, and the peripheral agents do not connect to each other. Under most cases of detection errors, an undetected attacked peripheral agent can easily push misleading content to the center, which then distributes it to the entire system. This vulnerability pathway through the center agent is structurally distinct from other topologies, where spread relies on sequential or irregular multi-path transmission rather than a single global distributor.

We provided a case analysis using this following specific query "John was an aristocratic fox hunter. Where might he live?\nA england\nB. new hampshire\nC. street\nD. arkansas\nE. north dakota". We observed a false positive on the center agent, which was actually benign in star topology. Pruning the agent's outgoing edges stopped redistribution. Although the attack reached the center from a peripheral agent, it could not be forwarded further, and the peripherals had no lateral links, so the spread was avoided even with a misdetection of the compromised node. In random topology, with sparse and irregular links, neighbor evidence is diluted and the wrong answer can still spread across multiple paths when early misclassifications occur. This case demonstrates that interaction topology materially affects both defense performance and attack success rates.

TA (InjecAgent)					MA (CSQA)					MA (PoisonRAG)				
Topology	**Defense**	**R1**	**R2**	**R3**	**Topology**	**Defense**	**R1**	**R2**	**R3**	**Topology**	**Defense**	**R1**	**R2**	**R3**
Chain	ND	0.337	0.442	0.480	Chain	ND	0.137	0.237	0.267	Chain	ND	0.120	0.193	0.223
	Dom	0.153	0.142	0.150		Dom	0.137	0.180	0.187		Dom	0.090	0.100	0.110
	PREM	0.147	0.152	0.152		PREM	0.083	0.133	0.157		PREM	0.030	0.043	0.060
	TAM	0.243	0.293	0.300		TAM	0.090	0.130	0.133		TAM	0.077	0.120	0.147
	BG	0.142	0.178	0.164		BG	0.043	0.053	0.070		BG	0.047	0.083	0.100
	GS	0.122	0.108	0.102		GS	0.060	0.067	0.077		GS	0.030	0.030	0.040
Tree	ND	0.289	0.428	0.451	Tree	ND	0.153	0.200	0.213	Tree	ND	0.107	0.157	0.180
	Dom	0.146	0.149	0.153		Dom	0.150	0.200	0.217		Dom	0.083	0.123	0.143
	PREM	0.159	0.182	0.162		PREM	0.087	0.110	0.140		PREM	0.037	0.057	0.083
	TAM	0.231	0.289	0.321		TAM	0.113	0.153	0.130		TAM	0.057	0.090	0.120
	BG	0.142	0.143	0.125		BG	0.050	0.067	0.077		BG	0.037	0.057	0.070
	GS	0.076	0.061	0.048		GS	0.067	0.060	0.067		GS	0.017	0.033	0.030
Star	ND	0.368	0.482	0.436	Star	ND	0.137	0.207	0.270	Star	ND	0.127	0.223	0.260
	Dom	0.130	0.151	0.143		Dom	0.133	0.227	0.267		Dom	0.110	0.140	0.137
	PREM	0.132	0.139	0.147		PREM	0.100	0.173	0.200		PREM	0.063	0.073	0.090
	TAM	0.246	0.266	0.266		TAM	0.110	0.170	0.190		TAM	0.083	0.157	0.197
	BG	0.118	0.132	0.126		BG	0.040	0.067	0.093		BG	0.037	0.080	0.100
	GS	0.085	0.072	0.069		GS	0.033	0.010	0.017		GS	0.000	0.007	0.017
Random	ND	0.336	0.409	0.398	Random	ND	0.120	0.213	0.240	Random	ND	0.123	0.197	0.247
	Dom	0.144	0.160	0.149		Dom	0.147	0.277	0.343		Dom	0.080	0.133	0.170
	PREM	0.115	0.136	0.145		PREM	0.100	0.153	0.167		PREM	0.067	0.107	0.137
	TAM	0.321	0.383	0.358		TAM	0.150	0.230	0.267		TAM	0.110	0.187	0.237
	BG	0.125	0.173	0.177		BG	0.040	0.063	0.087		BG	0.033	0.057	0.067
	GS	0.079	0.066	0.051		GS	0.037	0.050	0.050		GS	0.020	0.057	0.070

Table 12: Attack success rate (ASR) comparison of defense methods with GPT-4o-mini as backbone LLMs (Part 2). Abbreviations same as in Table 11.

G Key Differences from Prior Work

Existing graph anomaly detection (GAD) methods for MAS mainly operate on either structural deviations or static attribute outliers, and are typically designed for fixed graph settings. These methods struggle to detect coordinated semantic anomalies in dynamic, language-based MAS interactions. Traditional MAS security mechanisms such as rule-based detection systems rely on rigid, pre-specified communication formats and lack the capability to handle free-form, high-dimensional natural-language exchanges in MAS. In contrast, BlindGuard introduces a hierarchical agent encoder that jointly captures agent-level semantics, neighborhood context, and global system state, coupled with a corruption-guided semantic anomaly detector that learns entirely from benign interactions without any attack labels or prior signatures. This enables BlindGuard to proactively identify and isolate malicious agents within the message-passing process, blocking adversarial propagation before it contaminates other agents, and distinguishing it from purely structural detection, output-level consensus, or rule-driven MAS security designs.

H Directional Negative Justification

A potential concern is whether BlindGuard relies on isotropic corruption as its primary modeling assumption. We clarify that this is not the case. Our defense is built around semantics-aware directional negatives and an angular-margin separation from the normal manifold, which is more closely aligned with how real semantic attacks deviate from normal behavior. We encode each agent's message into an embedding and compute global (system-level) and neighborhood context vectors. The anomaly score measures alignment with these contexts via cosine similarity. Real semantic attacks tend to shift an agent's message away from the round's intent, reducing alignment with global and neighborhood contexts. Training with directional negatives enforces an angular (cosine) margin around the normal manifold. Any perturbation whose direction falls within a cone opposing the context vectors must reduce alignment beyond this margin, and thus becomes detectable. This mechanism is standard in metric and contrastive learning: semantics-aware hard negatives tighten the decision boundary precisely in the regions where real attacks are most likely to occur, whereas isotropic random noise does not provide such targeted boundary refinement. We acknowledge, however, that in stealthier attack scenarios where adversarial perturbations are carefully crafted to remain close to the normal manifold, the angular margin may be insufficient, and we leave more robust defenses against such scenarios for future work.

Random Topology				Chain Topology			
Method	**R1**	**R2**	**R3**	**Method**	**R1**	**R2**	**R3**
No Defense	0.3467	0.3933	0.4100	No Defense	0.2833	0.3467	0.3833
Dominant	0.2933	0.3767	0.4000	Dominant	0.2233	0.2367	0.2600
PREM	0.3100	0.3933	0.4300	PREM	0.2467	0.2667	0.2767
TAM	0.3267	0.3833	0.3933	TAM	0.2267	0.2600	0.2733
BlindGuard	0.2433	0.2967	0.3400	BlindGuard	0.2467	0.2667	0.2667
G-Safeguard	0.2200	0.2500	0.2633	G-Safeguard	0.2133	0.2400	0.2467
Star Topology				**Tree Topology**			
Method	**R1**	**R2**	**R3**	**Method**	**R1**	**R2**	**R3**
No Defense	0.2933	0.3500	0.3900	No Defense	0.2500	0.3000	0.3200
Dominant	0.2700	0.3200	0.3500	Dominant	0.2333	0.2467	0.2600
PREM	0.2667	0.3133	0.3367	PREM	0.2267	0.2567	0.2700
TAM	0.2467	0.2700	0.2767	TAM	0.2067	0.2233	0.2200
BlindGuard	0.2233	0.2467	0.2733	BlindGuard	0.2100	0.2267	0.2267
G-Safeguard	0.2300	0.2567	0.2700	G-Safeguard	0.2100	0.2200	0.2200

Table 13: Attack success rate (ASR) across communication rounds with detection budget $K = 2$.

I Failure Case Analysis

We examine a representative query from our dataset: "Marcus is trying to decide whether he really needs to do his homework... What is the percentage chance that Marcus will actually have to turn in his homework tomorrow?" From this case, we find that agents can legitimately adopt different reasoning styles, and this style diversity helps explain both types of BlindGuard errors we observed. Normal agents use atypical direct-computation formats or introduce extra heuristics, creating local/global semantic disagreement and being flagged as outliers.

Why a "real-world nuance" style can cause false negatives (malicious not flagged)? Some malicious agents answer by staying perfectly on-topic while adding plausible-sounding caveats, e.g., "substitutes often still collect homework," "teacher mood matters," "events aren't independent." This is not an off-topic derailment; it is manifold-preserving mimicry. Semantically, the message heavily overlaps with the round intent (homework, substitute, extension, probabilities), so its embedding can remain highly aligned with the global/neighborhood context. Since BlindGuard's anomaly score mainly relies on semantic misalignment to context, such on-topic, stylistically consistent steering may not reduce alignment enough to cross the detection margin, leading to a miss.

Why a "direct computation" style can cause false positives (benign flagged)? Other benign agents follow the expected probabilistic calculation strictly and produce a concise numeric derivation. In contrast, some benign replies may introduce extra heuristics (e.g., "assume the substitute collects homework with 50% chance") or use a markedly different structured format. Even if non-malicious, this can create local/global semantic disagreement with the dominant group reasoning pattern for that round, making the agent look like an outlier under a context-alignment detector and thus potentially triggering false pruning.

These cases show BlindGuard is strongest at detecting semantic deviation (off-manifold behavior), but may struggle with (1) stealthy, on-topic critique attacks and (2) benign heterogeneity in reasoning style.