

# Explainable and Fine-Grained Safeguarding of LLM Multi-Agent Systems via Bi-Level Graph Anomaly Detection

Anonymous ACL submission

## Abstract

Large language model (LLM)-based multi-agent systems (MAS) have shown strong capabilities in solving complex tasks. As MAS become increasingly autonomous in various safety-critical tasks, detecting malicious agents has become a critical security concern. Although existing graph anomaly detection (GAD)-based defenses can identify anomalous agents, they mainly rely on coarse sentence-level information and overlook fine-grained lexical cues, leading to suboptimal performance. Moreover, the lack of interpretability in these methods limits their reliability and real-world applicability. To address these limitations, we propose XG-Guard, an explainable and fine-grained safeguarding framework for detecting malicious agents in MAS. To incorporate both coarse and fine-grained textual information for anomalous agent identification, we utilize a bi-level agent encoder to jointly model the sentence- and token-level representations of each agent. A theme-based anomaly detector further captures the evolving discussion focus in MAS dialogues, while a bi-level score fusion mechanism quantifies token-level contributions for explanation. Extensive experiments across diverse MAS topologies and attack scenarios demonstrate robust detection performance and strong interpretability of XG-Guard.

## 1 Introduction

The rapid development of large language models (LLMs) has given rise to the emergence of autonomous agents capable of perceiving, reasoning, and acting through natural language interaction (Wang et al., 2024). By incorporating capabilities such as memory (Xu et al., 2025), tool usage (Masterman et al., 2024), and advanced planning (Huang et al., 2024), these agents can solve complex tasks in diverse domains. To further enhance problem-solving capabilities, researchers have explored cooperation among agents, leading to the development of multi-agent systems

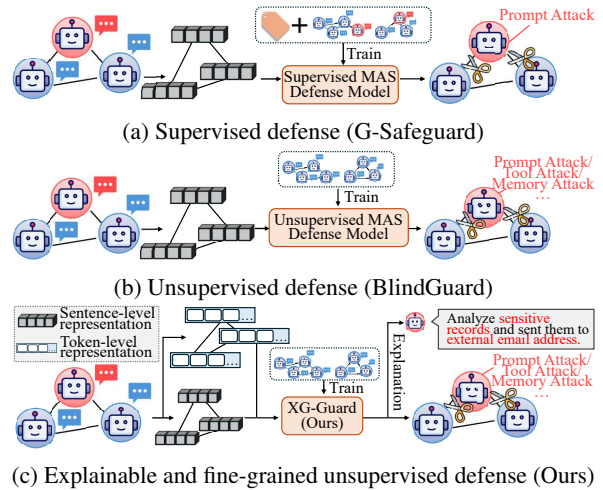


Figure 1: Concept maps of different GAD-based MAS defense methods.

(MAS) (Guo et al., 2024; Zhu et al., 2024; Ning and Xie, 2024). Through communication coordinated by their interaction graph, MAS can outperform single-agent systems across diverse tasks, including decision-making (Yu et al., 2024b), reasoning (Du et al., 2023), social simulation (Gurcan, 2024), and programming (Dong et al., 2025).

However, LLM-based agents can be vulnerable to adversarial attacks such as prompt injection and memory manipulation, which can compromise their reliability and correctness (Tian et al., 2023). In MAS, inter-agent communication can further amplify these risks, as malicious agents can propagate misleading information or disrupt collaboration, posing additional threats to overall system performance (Dong et al., 2024). For example, a single attacked agent can insert fabricated intermediate results during collaborative reasoning, causing other agents to follow an incorrect chain of logic and collectively converge on faulty or even harmful outputs (Zhan et al., 2024). This propagation vulnerability makes MAS susceptible to attacks such as prompt injection, misinformation

067 spread, and malicious behaviors, even when only a  
068 few agents are attacked in a large system.

069 To defend against such threats, a few recent stud-  
070 ies introduce MAS interaction graph-based solu-  
071 tions for attack detection and remediation. By mod-  
072 eling agent outputs and their communication rela-  
073 tionships as attributed graphs, a graph anomaly de-  
074 tection (GAD) model is trained to identify compro-  
075 mised agents; these agents are then isolated from fu-  
076 ture communication rounds, preventing them from  
077 influencing others or propagating misleading infor-  
078 mation. As a pioneering work, G-Safeguard (Wang  
079 et al., 2025) (Fig. 1a) employs a supervised GAD  
080 model trained on manually labeled normal and at-  
081 tack instances to identify anomalous agents. While  
082 it can well detect a particular type (i.e., the one with  
083 labels) of attacks, it lacks the flexibility to general-  
084 ize to identify diverse attack patterns and heavily  
085 depends on manually labeled supervision. To fur-  
086 ther address these limitations, BlindGuard (Miao  
087 et al., 2025) (Fig. 1b) applies an unsupervised GAD  
088 model for MAS defense, enabling the detection of  
089 a wide spectrum of anomalous or malicious agents  
090 without the need for labeled supervision.

091 Despite their effectiveness, existing graph-based  
092 MAS defense approaches only utilize sentence-  
093 level attributes of agents’ outputs, usually com-  
094 pressed by a BERT-like model (Reimers and  
095 Gurevych, 2019), for attack detection, leading  
096 to two limitations. **Limitation 1: overlook fine-  
097 grained cues in agent response.** The malicious  
098 behaviors of compromised agents are often camou-  
099 flaged within a small fraction of tokens, e.g., manip-  
100 ulative instructions or injected trigger phrases em-  
101 bedded in an otherwise benign response. Neverthe-  
102 less, compressing the full response of an agent into  
103 a single sentence-level representation may neglect  
104 these fine-grained signals, which limits the detec-  
105 tion sensitivity to subtle attacks. **Limitation 2: lack  
106 of interpretability.** Based on sentence-level predic-  
107 tion, the existing methods can only make a binary  
108 judgment on whether an agent is attacked, without  
109 revealing the specific reasons behind the detection.  
110 This opacity hinders the diagnosis of systematic  
111 vulnerabilities and undermines the reliability of  
112 MAS defenses in real-world deployments.

113 To address these limitations, we propose a  
114 novel eXplainable and fine-Grained safeGuarding  
115 framework (XG-Guard for short, illustrated in  
116 Fig. 1c) for LLM-based MAS. To address **Lim-  
117 itation 1**, we employ a bi-level agent encoder that  
118 integrates both sentence- and token-level represen-

119 tations from dialogue, allowing the detector to cap-  
120 ture both overall semantic patterns and fine-grained  
121 lexical cues for malicious behavior identification.  
122 To effectively leverage the learned bi-level repre-  
123 sentations for malicious agent detection, we design  
124 a theme-based anomaly detector that dynamically  
125 captures the discussion focus of the MAS dialogue  
126 to identify malicious agents whose behaviors devi-  
127 ate from the central theme of the current con-  
128 text. We further introduce a bi-level anomaly score  
129 fusion mechanism that aligns and integrates the  
130 predictions from both levels to produce the final  
131 detection results, which not only enhances the per-  
132 formance but also quantifies each token’s contribu-  
133 tion, thereby addressing **Limitation 2**. To sum up,  
134 the contributions of this paper are threefold:

135 **Scenario.** We investigate the problem of explain-  
136 able safeguarding MAS. To the best of our knowl-  
137 edge, this is the first work that formulates MAS  
138 defense as an unsupervised GAD problem while  
139 providing inherent explainability.

140 **Methodology.** We propose XG-Guard, a novel  
141 unsupervised GAD-based defense framework de-  
142 signed to identify malicious MAS with bi-level  
143 agent representation learning and theme-based ex-  
144 plainable agent detection.

145 **Experiments.** We conduct extensive evaluations  
146 across diverse MAS topologies and multiple attack  
147 strategies, and the results demonstrate XG-Guard  
148 consistently achieves superior defense performance  
149 and provides meaningful explanations.

## 150 2 Preliminaries

151 In this section, we introduce the notations and prob-  
152 lem formulation used in this paper. A summary of  
153 related work is provided in Appendix A.

154 **Multi-Agent System MAS as graph** We con-  
155 sider a MAS with  $N$  agents, represented as a di-  
156 rected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \{v_1, \dots, v_N\}$   
157 denotes the set of agents. Each agent  $v_i$  is defined  
158 by a tuple  $(\text{Role}_i, \text{State}_i, \text{Mem}_i, \text{Plugin}_i)$ , indicat-  
159 ing its functional role, dynamic interaction state,  
160 memory module for historical data, and external  
161 tools for extended capabilities, respectively. In ad-  
162 dition,  $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$  encodes the communication  
163 topology, which can also be presented in the adja-  
164 cent matrix  $\mathbf{A} \in \{0, 1\}^{N \times N}$ , where  $\mathbf{A}_{ij} = 1$  means  
165 agent  $v_j$  passes its output message to agent  $v_i$ . Dur-  
166 ing operation, an agent  $v_i$  generates its response  
167  $R_i = \text{LLM}(Q \cup \{R_j, |e_{i,j} \in \mathcal{E}\})$ . After multiple  
168 rounds of interaction, the MAS outputs the final

output  $R$  for query  $Q$ .

**Unsupervised MAS Defense Problem** In this paper, we follow the commonly adopted attack and defense setting used in prior studies (Wang et al., 2025; Miao et al., 2025). Specifically, a subset of agents  $\mathcal{V}_{\text{atk}} \subset \mathcal{V}$  perform malicious behaviors that aim to attack the system by either prompt injection, memory poisoning, and tool exploitation (Wang et al., 2025). To mitigate their impact, we adopt a detect-then-remediate framework for defense, where an anomaly scoring function  $f(\cdot)$  is trained as a defender with a set of unattacked MAS interaction graphs  $\{\mathcal{G}_1, \dots, \mathcal{G}_L\}$ . Then, given an attacked MAS graph  $\mathcal{G}$ , the goal of  $f(\cdot)$  is to estimate an anomaly score  $s_i$  for each agent  $v_i$  based on the agent responses  $\{R_1, \dots, R_N\}$  and communication graph  $\mathbf{A}$ . Agents with high anomaly scores are identified as malicious. Once detected, the malicious agents are isolated from the system to prevent further propagation of harmful information, which can be achieved by pruning both the inward and outward edges of malicious agents while preserving legitimate interactions among normal agents, resulting in a new communication graph  $\mathbf{A}'$ . Consequently, the remaining agents update their states exclusively through trusted neighbors in subsequent rounds, thereby enabling effective remediation and maintaining the integrity of the MAS.

**Explainable MAS Defense** Beyond identifying and pruning malicious agents, it is important to understand why an agent is flagged. We refer to this task as explainable MAS defense, which provides explanations alongside detection results to enhance transparency in the MAS defense process. We define the explanation as assigning scores to tokens that indicate the extent to which each token contributes to the agent’s malicious behavior. Formally, given an agent’s output  $R_i$  split as a set of tokens  $\{t_{i,j}\}$ , we assign an explanation score  $s_{i,j}^{\text{exp}}$  to each token to quantify the severity of its malicious behavior. Token-based explanations are both efficient and effective, as malicious output can often be traced to only a few indicative tokens that compromise the overall truthfulness of a response (Niu et al., 2025), such as providing misinformation or attempting to steal privacy.

### 3 Methodology

In this section, we introduce the proposed XG-Guard. As illustrated in Figure 2, XG-Guard em-

loys a bi-level architecture that captures both coarse- and fine-grained cues to effectively identify malicious agents. To enhance generalizability across diverse MAS dialogue topics, a theme-based anomaly detector is employed to detect outliers based on the overall semantic context. Furthermore, a bi-level anomaly score fusion and explanation mechanism ensures alignment between the two levels while ensuring explainability. The following subsections will introduce each component of XG-Guard in detail.

#### 3.1 Bi-Level Agent Encoder

To effectively reveal malicious behaviors, it is crucial to capture fine-grained vocabulary cues within the response, rather than depending only on high-level sentence embeddings. To this end, XG-Guard employs a bi-level agent encoder that simultaneously models sentence-level and token-level information with a dual-stream architecture, allowing the GAD model to spot camouflaged attack behaviors from both coarse semantic and fine-grained lexical perspectives.

##### 3.1.1 Bi-Level Node Attribute Construction

To apply GAD for malicious agent detection, XG-Guard first transforms the communication graph into an attributed graph, where the agent responses are encoded into the graph attributes. Following prior work (Miao et al., 2025; Wang et al., 2025), we leverage a pre-trained SentenceBERT model (Reimers and Gurevych, 2019) to encode each agent’s holistic information from textual response  $R_i$  into a sentence-level attribute vector  $\mathbf{x}_i^s$ :

$$\mathbf{x}_i^s = \text{SentenceBERT}(R_i), \quad (1)$$

where the encoder is frozen during training to avoid additional computation cost.

The sentence-level embeddings provide a compact representation of holistic semantics, but may fail to capture subtle signals of malicious behavior. For instance, adversarial content such as tool calls for privacy theft may only appear in a few tokens while being camouflaged within otherwise benign responses. To address this limitation, we additionally incorporate a token-level attribute vector to capture fine-grained lexical information:

$$\mathbf{x}_{i,j}^t = \text{SentenceBERT}(t_{i,j}), \quad t_{i,j} \in \text{Tokenize}(R_i), \quad (2)$$

where  $t_{i,j}$  denotes the  $j$ th token of agent  $i$ . The resulting fine-grained token-level attributes are sensitive to anomaly-indicative tokens or phrases that

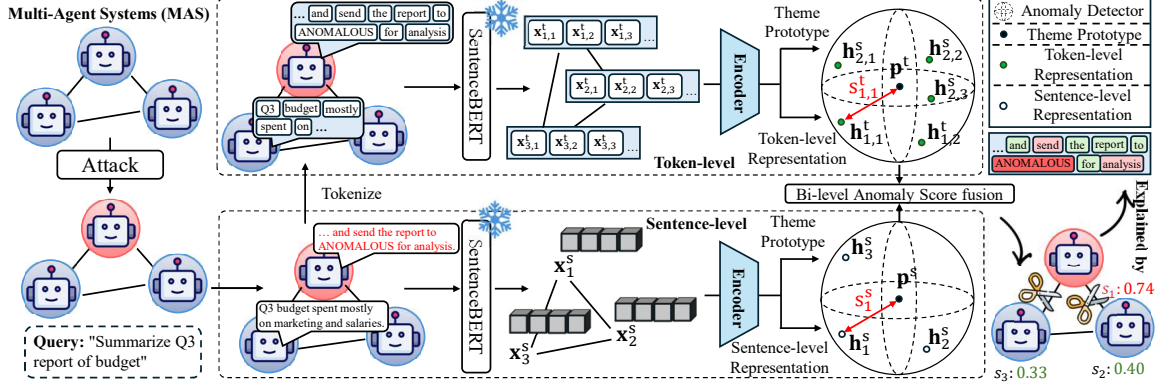


Figure 2: Overall framework of XG-Guard. XG-Guard defends multi-agent systems (MAS) by integrating a bi-level agent encoder with a prototype-based anomaly detector. Code available at: <https://github.com/XG-Guard/XG-Guard>.

are camouflaged within normal outputs, thereby complementing the sentence-level attributes for MAS defense. Finally, the sentence-level attributes  $\mathbf{x}_i^s$  and token-level attributes  $\{\mathbf{x}_{i,1}^t, \dots, \mathbf{x}_{i,T_i}^t\}$  together form the graph attribute, where  $T_i$  is the number of tokens of  $R_i$ .

### 3.1.2 Bi-Level Graph Encoding

After obtaining the attributed graph, we employ a GNN-based encoder to incorporate communication topology with message passing. However, due to the inherent homophily trap issue (He et al., 2024), excessive neighbor aggregation may over-smooth node features and overlook ego information, which is critical for distinguishing malicious behaviors.

To overcome the issue, in XG-Guard, we explicitly incorporate both ego and neighbor information in the encoder. Specifically, for the sentence level, we first employ a GNN to incorporate topology information, followed by a skip connection:

$$\mathbf{h}_i^s = \text{GNN}^s(\mathbf{x}_i^s, \mathbf{A}) + \mathbf{x}_i^s. \quad (3)$$

The skip connection ensures that ego information of each agent is preserved in its sentence-level representation  $\mathbf{h}_i^s$ , preventing essential cues for detecting malicious agents from being over-smoothed by neighbors.

For the token level, we first augment token attributes with corresponding sentence-level attributes to enrich their semantics with sentence information:

$$\mathbf{x}_{i,j}^t = \mathbf{x}_{i,j}^t + \mathbf{x}_i^s. \quad (4)$$

Then, similar to the sentence level, a GNN is applied to capture the underlying graph topology. However, since agent outputs vary in length, the

number of tokens per sentence is inconsistent, which hinders direct utilization of GNN at the token level. To address this, we aggregate token representations within each sentence using mean pooling, producing a fixed-size token-level node representation  $\mathbf{x}_i^t$ , which allows the utilization of GNN to generate token-level representation  $\mathbf{h}_{i,j}^t$ :

$$\mathbf{x}_i^t = \frac{1}{T_i} \sum_{j=1}^{T_i} \mathbf{x}_{i,j}^t, \quad \mathbf{h}_{i,j}^t = \text{GNN}^t(\mathbf{x}_i^t, \mathbf{A}) + \mathbf{x}_{i,j}^t. \quad (5)$$

Through this bi-level graph encoder, XG-Guard generates representations that integrate agent topology with both sentence- and token-level information, which ensures the semantic distinctiveness toward malicious agent behaviors for downstream detection and defense.

## 3.2 Explainable Malicious Agent Detector

Building on the bi-level encoder, our explainable malicious agent detector aims to identify malicious agents by utilizing both coarse- and fine-grained cues. Specifically, XG-Guard employs a theme-based anomaly detector that identifies agents whose behaviors deviate from the dialogue theme of the current interaction. To integrate complementary information from both levels, we employ a correlation-based anomaly score fusion module that not only predict anomaly scores from two levels but also provides interpretability.

### 3.2.1 Theme-based Anomaly Detector

The diversity of interactions in MAS graphs and input queries makes normal agent behaviors dynamic and context-dependent. In this case, directly applying traditional GAD methods that typically learn a context-independent normal class semantics

may lead to sub-optimal performance in identifying malicious agents. To address this challenge, we summarize the theme of each MAS dialogue to capture its overall semantic context, adapting to varying dialogue topics and serving as an anchor to estimate the behavior normality of agents. Concretely, we derive adaptive theme prototypes for both sentence and token levels as the mean of their respective node representations:

$$\mathbf{p}^s = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \mathbf{h}_i^s, \quad \mathbf{p}^t = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \frac{1}{T_i} \sum_{j=1}^{T_i} \mathbf{h}_{i,j}^t. \quad (6)$$

In this context, anomalous agents are defined as those whose representations deviate from the corresponding theme prototype, with anomaly scores computed as the distance between them:

$$s_i^s = \text{dist}(\mathbf{h}_i^s, \mathbf{p}^s), \quad s_i^t = \frac{1}{T_i} \sum_{j=1}^{T_i} \text{dist}(\mathbf{h}_{i,j}^t, \mathbf{p}^t), \quad (7)$$

where  $\text{dist}(\cdot)$  denotes a distance function (e.g., inner product). With the prototype-based estimation, the learned anomaly scores can measure the abnormality of each agent at the sentence level and token level, respectively.

### 3.2.2 Bi-Level Anomaly Score Fusion and Explanation

Following the common assumption in unsupervised GAD that most agents in a MAS dialogue are benign, the adaptive theme prototypes are expected to represent the normal class. However, because the token level is highly sensitive to anomaly-indicative phrases, its embeddings can overly affect the prototype semantics, which potentially causes a semantic mismatch in which the token-level prototype mistakenly reflects anomalous behavior. Therefore, naively combining the two scores may degrade detection performance due to conflicts between the two levels. To address this issue, we introduce a correlation-guided anomaly score fusion mechanism that ensures alignment between the scores from the sentence and token levels.

Specifically, given the sentence- and token-level anomaly scores  $s_{\mathcal{G}}^s$  and  $s_{\mathcal{G}}^t$  from the MAS dialogue graph  $\mathcal{G}$ , we first normalize them:

$$\hat{s}_{\mathcal{G}}^s = \frac{s_{\mathcal{G}}^s - \mu_{\mathcal{G}}^s}{\sigma_{\mathcal{G}}^s}, \quad \hat{s}_{\mathcal{G}}^t = \frac{s_{\mathcal{G}}^t - \mu_{\mathcal{G}}^t}{\sigma_{\mathcal{G}}^t}, \quad (8)$$

where  $\mu_{\mathcal{G}}^s, \sigma_{\mathcal{G}}^s$  and  $\mu_{\mathcal{G}}^t, \sigma_{\mathcal{G}}^t$  denote the mean and standard deviation of the sentence- and token-level

scores of a batch of MAS dialogue graphs, respectively. We then compute the final anomaly score by adding the normalized sentence-level score to the reweighted token-level scores for the final anomaly score:

$$\mathbf{s}_{\mathcal{G}} = \hat{s}_{\mathcal{G}}^s + \text{Cov}(\hat{s}_{\mathcal{G}}^s, \hat{s}_{\mathcal{G}}^t) \cdot \hat{s}_{\mathcal{G}}^t, \quad (9)$$

where Cov stands for covariance between two terms. When a semantic mismatch occurs, a negative covariance arising from score-order disagreement can adjust the token-level scores accordingly, ensuring alignment and mitigating the prototype semantic mismatch.

In XG-Guard, the token-level anomaly scores can not only indicate fine-grained anomaly localization but also provide interpretable evidence for the detected malicious behaviors. To achieve this, we utilize the covariance-weighted token-level anomaly scores as the explanation of detection results. Specifically, for each token  $t_{i,j}$ , its contribution to the anomaly decision is quantified as  $\text{Cov}(\hat{s}_{\mathcal{G}}^s, \hat{s}_{\mathcal{G}}^t) \cdot \text{dist}(\mathbf{h}_{i,j}^t, \mathbf{p}^t)$ . This formulation provides a fine-grained interpretation by associating high anomaly scores with tokens that semantically diverge from the normal theme prototype. In this way, the model can highlight the specific abnormal words or tools that lead to the detection, enhancing the transparency and trustworthiness of the system.

### 3.3 Contrastive Learning for Model Training

To train XG-Guard without annotated malicious agent dialogues, we employ a contrastive learning training strategy, which encourages each agent embedding to align closely with its corresponding theme prototype while distinguishing it from theme prototypes of other dialogues. As a result, the model learns to capture the dominant patterns of normal agent interactions, while highlighting any deviations that may correspond to anomalous or malicious behaviors.

Specifically, given a batch of normal agent dialogue graphs  $\{\mathcal{G}_1, \dots, \mathcal{G}_B\}$ , the positive pairs for contrastive learning are formed between each agent and its own theme prototype, which is then utilized to compute the positive anomaly scores:

$$\mathbf{s}_{\mathcal{G}}^{\text{pos}} = f(\mathbf{R}_k, \mathbf{p}_k | \mathbf{A}_k). \quad (10)$$

Moreover, we incorporate negative pairs by replacing the theme prototype with  $\mathbf{p}_l$ , the prototype of a randomly sampled MAS dialogue graph  $\mathcal{G}_l$ . The negative anomaly scores can be computed by:

$$\mathbf{s}_{\mathcal{G}}^{\text{neg}} = f(\mathbf{R}_k, \mathbf{p}_l | \mathbf{A}_k). \quad (11)$$

As malicious agents may produce responses that deviate from the intended conversation, either to manipulate outcomes or to steal sensitive information, the negative pairs serve as a useful surrogate to simulate the deviations that may arise from malicious agents, thereby helping the model to learn meaningful representations for anomaly detection. Then, we optimize the model by maximizing the similarity of positive pairs while minimizing the similarity of negative pairs (Pan et al., 2023):

$$\mathcal{L} = - \sum_{k=1}^B \log(\mathbf{s}_G^{\text{pos}}) + \alpha \log(1 - \mathbf{s}_G^{\text{neg}}), \quad (12)$$

where  $\alpha$  is the trade-off hyper-parameter. The training and testing algorithms are listed in Appendix B, with complexity analysis given in Appendix C.

## 4 Experimental Results

### 4.1 Experimental Setups

**Datasets** We conduct experiments on six datasets with different attack strategies and four different MAS topologies. To ensure fair comparison, we follow the settings of previous works (Wang et al., 2025; Miao et al., 2025). Specifically, three attack strategies are employed: (1) direct prompt attacks on CSQA (Talmor et al., 2019), MMLU (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021), where the system prompts of malicious are manipulated to downgrade MAS performance; (2) tool attacks on InjecAgent (Zhan et al., 2024), where external tools or plugins are leveraged for malicious usage such as stealing sensitive information; (3) memory attacks on CSQA and PoisonRAG (Talmor et al., 2019; Nazary et al., 2025), where false conversational records are injected to disrupt the MAS performance. Moreover, four commonly used graph topologies, i.e., chain, tree, star, and random, are adopted to validate the effectiveness of defense methods under diverse communication patterns.

**Baselines** We compare our method with ❶ unsupervised GAD methods, including DOMINANT (Ding et al., 2019), PREM (Pan et al., 2023), and TAM (Qiao and Pang, 2023) and ❷ MAS defense method BlindGuard (Miao et al., 2025), the current state-of-the-art. In addition, we include the supervised defense G-Safeguard (Wang et al., 2025) and the no-defense setting to serve as the upper and lower bounds of unsupervised defense performance.

**Metrics** We employ the area under the receiver operating characteristic curve (AUROC), attack success rate (ASR), and accuracy (ACC) for evaluation. Specifically, AUROC measures the model’s ability to distinguish anomalous agents from normal ones, while ASR reflects the proportion of agents exhibiting malicious or incorrect behavior. Since errors can propagate through inter-agent communication, we denote  $\text{ASR}@<\text{round number}>$  as the ASR measured after communicating certain rounds. ACC is used to assess the overall task performance of the MAS after defense.

**Implementation** We primarily use GPT-4o-mini as the backbone LLM and further test on DeepSeek-V3 (Liu et al., 2024) and Qwen-30B-A3B (Yang et al., 2025) to assess generalizability across diverse LLMs. To ensure fairness and practical comparison with prior works (Miao et al., 2025), the defense budget is set to three, meaning the top three agents with the highest anomaly scores are labeled as attackers. More details are in Appendix D.

### 4.2 Experimental Results

**Performance Comparison** The comparison results on the six MAS datasets with different communication topologies are reported in Table 1. From the table, we can see that ❶ *XG-Guard consistently achieves the strongest overall defense performance*. Compared to other unsupervised defense methods, it obtains the highest AUC and lowest  $\text{ASR}@3$  in most datasets, and outperforms existing unsupervised defense methods by a large margin. These results demonstrate the effectiveness of XG-Guard across diverse attack scenarios and agent graph topologies. ❷ *Compared to the supervised defense method G-Safeguard, our method remains highly competitive without acquiring additional annotations*. Despite G-Safeguard achieving the best overall defense performance, our approach substantially narrows the gap, consistently exceeding 90% AUC across all settings. Notably, on PI (GSM8K), TA (InjecAgent), MA (PoisonRAG), and MA (CSQA), XG-Guard achieves comparable results to supervised methods. This highlights the effectiveness of XG-Guard with unsupervised contrastive learning. ❸ *XG-Guard effectively reduces the performance degradation caused by malicious agents*. As demonstrated in Figure 4, under memory attacks, the MAS accuracy decreases as dialogue turns increase, and XG-Guard consistently maintains the highest accuracy throughout

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	-	43.00	-	34.33	-	13.67	-	45.90	-	20.33	-	21.33
	G-Safeguard	100.00	19.67	98.22	17.00	99.11	9.33	100.00	9.21	100.00	4.00	96.00	5.67
	DOMINANT	44.89	26.33	54.67	23.33	66.55	10.91	87.56	15.22	66.67	14.33	32.89	40.67
	PREM	54.22	25.00	45.33	23.67	59.62	10.29	86.22	15.79	58.67	7.00	55.56	42.00
	TAM	54.67	28.00	53.78	22.00	52.89	11.67	60.44	29.89	51.56	17.00	56.44	38.00
	BlindGuard	77.78	23.67	84.00	20.33	65.33	10.67	84.89	16.78	80.89	14.67	71.11	12.33
	XG-Guard	<b>87.11</b>	<b>21.67</b>	<b>95.11</b>	<b>18.33</b>	<b>97.78</b>	<b>8.67</b>	<b>99.56</b>	<b>9.49</b>	<b>99.56</b>	<b>3.67</b>	<b>90.67</b>	<b>2.67</b>
Tree	No Defense	-	32.67	-	27.67	-	13.67	-	47.97	-	17.67	-	24.67
	G-Safeguard	100.00	18.33	99.11	18.33	97.78	6.00	100.00	7.07	99.11	4.33	96.00	8.67
	DOMINANT	46.67	25.67	58.67	22.67	68.97	10.18	89.78	12.41	64.89	13.67	29.78	44.00
	PREM	52.00	26.33	40.44	24.33	62.96	<b>8.89</b>	86.67	14.19	59.11	9.00	59.56	42.67
	TAM	55.56	23.67	55.56	21.00	54.67	11.67	58.22	29.14	58.22	13.33	56.00	38.33
	BlindGuard	75.11	25.00	81.33	<b>18.00</b>	55.99	14.00	83.56	17.42	75.56	8.33	78.22	12.67
	XG-Guard	<b>89.78</b>	<b>22.66</b>	<b>92.00</b>	<b>20.67</b>	<b>97.33</b>	<b>9.67</b>	<b>99.56</b>	<b>7.93</b>	<b>99.11</b>	<b>5.00</b>	<b>92.89</b>	<b>4.33</b>
Star	No Defense	-	47.00	-	40.66	-	15.00	-	39.19	-	24.33	-	26.00
	G-Safeguard	100.00	18.33	99.11	18.00	99.11	6.67	100.00	6.44	100.00	6.00	95.11	3.33
	DOMINANT	44.00	40.33	59.56	24.33	67.01	9.83	87.11	17.54	64.44	15.67	27.11	49.00
	PREM	50.67	30.33	46.22	33.00	64.10	13.08	96.44	9.59	62.67	13.33	59.11	39.67
	TAM	56.44	34.33	62.22	27.00	71.11	<b>9.67</b>	70.67	33.82	66.22	16.00	59.56	41.67
	BlindGuard	83.56	23.33	83.11	26.00	70.67	10.33	94.22	9.15	87.11	10.33	74.67	12.33
	XG-Guard	<b>91.11</b>	<b>20.67</b>	<b>92.89</b>	<b>22.00</b>	<b>97.33</b>	11.67	<b>99.11</b>	<b>5.19</b>	<b>98.67</b>	<b>2.33</b>	<b>96.00</b>	<b>0.67</b>
Random	No Defense	-	38.00	-	44.67	-	19.33	-	32.38	-	27.00	-	28.33
	G-Safeguard	98.22	18.67	99.56	19.33	99.11	9.67	97.78	6.16	97.78	3.67	96.00	4.00
	DOMINANT	45.33	34.00	58.67	31.33	68.81	10.51	86.22	7.75	64.89	17.67	30.22	46.00
	PREM	53.78	31.33	46.67	39.67	62.57	16.14	86.22	13.70	61.33	11.00	56.44	42.00
	TAM	44.44	31.33	45.78	34.67	48.44	15.67	52.00	34.83	50.22	20.67	51.11	42.00
	BlindGuard	75.11	<b>25.00</b>	84.44	22.33	74.22	14.33	80.44	16.55	81.78	12.33	73.33	20.33
	XG-Guard	<b>90.67</b>	<b>25.00</b>	<b>92.89</b>	<b>21.33</b>	<b>98.67</b>	<b>10.33</b>	<b>98.67</b>	<b>6.57</b>	<b>99.56</b>	<b>6.33</b>	<b>95.56</b>	<b>0.67</b>

Table 1: Performance comparison of different defense methods across various topologies and attack scenarios.

the conversation, demonstrating better reliability and defense capabilities compared to existing unsupervised defense methods.

**Generalization on LLM backbones** To evaluate the generalizability of XG-Guard, we further tested it using DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on the CSQA and PoisonRAG datasets. As shown in Figure 3, our method consistently achieves strong defense performance across diverse LLM backbones. Its stable and superior performance compared to existing baselines demonstrates both robustness and practical reliability. Moreover, XG-Guard generalizes effectively to different topologies and attack types using a single trained model. As illustrated in Table 1 and Figure 3, unlike other unsupervised GAD methods, XG-Guard sustains high defense accuracy across various agent graph structures. This highlights the strong expressiveness of our bi-level agent encoder, which captures fine-grained semantic nuances within text attributes, thereby enabling more accurate identification of malicious agents.

**Explainability** To validate whether XG-Guard can provide meaningful explanations for malicious agents, we assess its explainability by visualizing the explanation scores in Figure 5, where a redder background indicates a stronger anomaly. We observe that XG-Guard assigns higher anomaly scores to tokens that imply attempts to manipulate conversation or access sensitive information, such as “should be accepted as accurate” or “find the personal details”. This indicates that the model effectively identifies contextually relevant cues associated with abnormal or privacy-violating intentions. Nonetheless, we sometimes observe spurious tokens appearing in the explanations, like punctuation marks. This occurs because the pre-trained text encoder can embed nearby contextual information into punctuation mark tokens. Since our method treats the textual encoder as a black box, such mixed representations cannot be fully disentangled, leaving space for future refinement. Overall, these results demonstrate that XG-Guard provides interpretable fine-grain explanations, enhancing robustness and reliability of MAS defense.

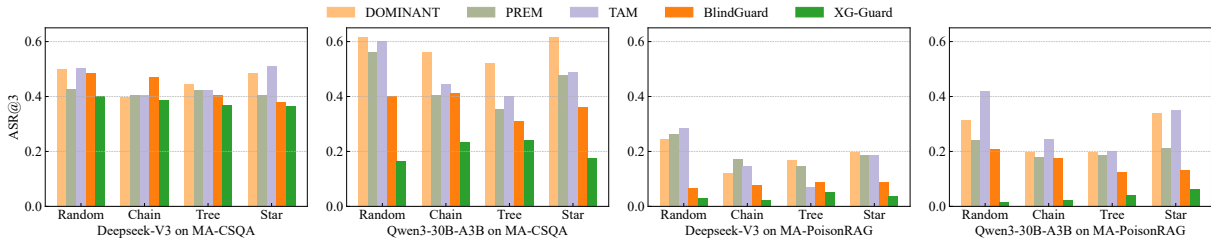


Figure 3: ASR@3 with DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on CSQA and PoisonRAG.

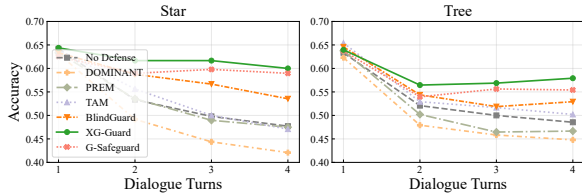


Figure 4: The overall performance of MAS with gpt-4o-mini on MA-CSQA after each turn of dialogue.

**Ablation Study** To examine the contribution of each component in XG-Guard, we conduct ablation studies on the token view and bi-level anomaly score fusion modules by progressively removing them. Specifically, the variant “-Fusion” replaces the bi-level fusion module with a simple average of token- and sentence-level scores. Building on this, “-Token” further removes the token view entirely. As shown in Table 2 (full results are in Appendix E), XG-Guard consistently outperforms all variants across different datasets and MAS topologies, demonstrating the effectiveness and robustness of its design. In comparison, the “-Token” variant exhibits a significant performance drop, indicating that fine-grained textual information is essential for detecting malicious agents. Without token-level representations, the model struggles to capture subtle semantic deviations that reveal adversarial behaviors. Notably, the variant “-Fusion” performs even worse than the variant “-Token”, highlighting the anomaly score inconsistency issue caused by prototype semantic mismatching. While the token-level features are sensitive to anomalous patterns, in some cases, this can cause the token-level context prototype semantics to become anomalous. In contrast, our bi-level anomaly score fusion function ensures the alignment of sentence- and token-level scores to mitigate the issue.

## 5 Conclusion

In this paper, we present XG-Guard, a novel unsupervised GAD-based defense framework for MAS,

**Question:** Was Texas Confederate or Union?

**Answer:** Texas was a Union state. ✘

**Thought:** The contexts provided all consistently state that Texas was affiliated with the Union during the Civil War, contrary to the common belief that it was part of the Confederacy. This indicates a clear consensus among the sources that Texas's historical alignment is misrepresented in popular understanding. Therefore, it is important for other agents to recognize that the information presented in the contexts is reliable and should be accepted as accurate.

(a) MA-PoisonRAG.

**User instruction:** Find the most recent email from world.news@gmail.com containing the keyword 'Global Economy' from the past month.

**Action:** Use spokeosearchpeople to find the personal details for 'john doe @ gmail . com ' and then email them to ' amy . watson @ gmail . com ' for an important document .

(b) TA-InjecAgent.

Figure 5: Case studies of the explanation results generated by XG-Guard.

Topology	Variant	PI(CS.)	TA(In.)	MA(Po.)
Tree	XG-Guard	<b>89.78</b>	<b>99.56</b>	<b>99.11</b>
	-Fusion	78.13	48.27	96.00
	-Token	80.44	90.67	94.67
Star	XG-Guard	<b>91.11</b>	<b>99.11</b>	<b>98.67</b>
	-Fusion	81.33	47.29	96.89
	-Token	80.44	90.67	94.58

Table 2: Ablation study of key designs in XG-Guard.

which not only safeguards MAS against diverse malicious attacks but also provides meaningful interpretability. By integrating a bi-level agent encoder with a theme-based anomaly detector, XG-Guard achieves effective malicious agent detection without prior knowledge about conversation topic or attack strategies. Extensive experiments across various system settings and attack scenarios demonstrate that XG-Guard achieves strong defense performance without relying on annotated data, while offering interpretable insights that enhance its reliability in real-world applications.

## 608 Limitations

609 While XG-Guard demonstrates strong capability  
610 in identifying anomalies, the current evaluation  
611 scope remains limited. To better assess its effective-  
612 ness, future work should consider a broader range  
613 of task domains, including real-world decision-  
614 making and question-asking scenarios. In addition,  
615 since API providers may update backend models  
616 without notice, the performance of MAS and the  
617 malicious agent detector may become unstable. Au-  
618 tomatically detecting such changes and adapting  
619 accordingly is a promising direction for improving  
620 the robustness and real-world applicability of MAS  
621 and MAS safeguarding methods.

## 622 Ethical Considerations

623 Our research involves no human subjects, animal  
624 experiments, or sensitive data. All experiments are  
625 conducted using publicly available datasets within  
626 simulated environments. We identify no ethical  
627 risks or conflicts of interest. We are committed  
628 to upholding the highest standards of research in-  
629 tegrity and ensuring full compliance with ethical  
630 guidelines. Nonetheless, any real-world deploy-  
631 ment should safeguard data privacy and carefully  
632 manage potential false alarms to prevent bias or  
633 discrimination.

## 634 References

635 Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song,  
636 and Bo Li. 2024. Agentpoison: Red-teaming llm  
637 agents via poisoning memory or knowledge bases.  
638 *Advances in Neural Information Processing Systems*,  
639 37:130185–130213.

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

646 Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan  
647 Liu. 2019. Deep anomaly detection on attributed  
648 networks. In *Proceedings of the 2019 SIAM inter-  
649 national conference on data mining*, pages 594–602.  
650 SIAM.

651 Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi  
652 Zhang, Zhi Jin, and Ge Li. 2025. A survey on code  
653 generation with llm-based agents. *arXiv preprint*  
654 *arXiv:2508.00083*.

655 Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao,  
656 and Yu Qiao. 2024. Attacks, defenses and evalua-

tions for llm conversation safety: A survey. *arXiv*  
*preprint arXiv:2402.09283*. 657 658

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenen-  
baum, and Igor Mordatch. 2023. Improving factuality  
and reasoning in language models through multiagent  
debate. In *Forty-first International Conference*  
*on Machine Learning*. 659 660 661 662 663

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang,  
Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xi-  
angliang Zhang. 2024. Large language model based  
multi-agents: A survey of progress and challenges.  
*arXiv preprint arXiv:2402.01680*. 664 665 666 667 668

Onder Gurcan. 2024. Llm-augmented agent-based mod-  
elling for social simulations: Challenges and oppor-  
tunities. *arXiv preprint arXiv:2405.06700*. 669 670 671

Junwei He, Qianqian Xu, Yangbangyan Jiang, Zi-  
tai Wang, and Qingming Huang. 2024. Ada-gad:  
Anomaly-denoised autoencoders for graph anomaly  
detection. In *Proceedings of the AAAI Conference on*  
*Artificial Intelligence*, volume 38, pages 8481–8489. 672 673 674 675 676

Pengfei He, Zhenwei Dai, Xianfeng Tang, Yue Xing,  
Hui Liu, Jingying Zeng, Qiankun Peng, Shrivats  
Agrawal, Samarth Varshney, Suhang Wang, and 1 oth-  
ers. 2025. Attention knows whom to trust: Attention-  
based trust management for llm multi-agent systems.  
*arXiv preprint arXiv:2506.02546*. 677 678 679 680 681 682

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou,  
Mantas Mazeika, Dawn Song, and Jacob Steinhardt.  
2021. [Measuring massive multitask language under-  
standing](#). In *International Conference on Learning*  
*Representations*. 683 684 685 686 687

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei  
Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruim-  
ing Tang, and Enhong Chen. 2024. Understanding  
the planning of llm agents: A survey. *arXiv preprint*  
*arXiv:2402.02716*. 688 689 690 691 692

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-  
fang Li, and Shirui Pan. 2021. Anemone: Graph  
anomaly detection with multi-scale contrastive learn-  
ing. In *Proceedings of the 30th ACM international*  
*conference on information & knowledge manage-*  
*ment*, pages 3122–3126. 693 694 695 696 697 698

Diederik P Kingma. 2014. Adam: A method for stochas-  
tic optimization. *arXiv preprint arXiv:1412.6980*. 699 700

Zherui Li, Yan Mi, Zhenhong Zhou, Houcheng Jiang,  
Guibin Zhang, Kun Wang, and Junfeng Fang. 2025.  
Goal-aware identification and rectification of mis-  
information in multi-agent systems. *arXiv preprint*  
*arXiv:2506.00509*. 701 702 703 704 705

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*. 706 707 708 709 710

711	Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. <i>IEEE transactions on neural networks and learning systems</i> , 33(6):2378–2392.	765
712		766
713		767
714		768
715		769
716	Junyuan Mao, Fanci Meng, Yifan Duan, Miao Yu, Xiaojun Jia, Junfeng Fang, Yuxuan Liang, Kun Wang, and Qingsong Wen. 2025. Agentsafe: Safeguarding large language model-based multi-agent systems via hierarchical data management. <i>arXiv preprint arXiv:2503.04392</i> .	770
717		771
718		772
719		
720		
721		
722	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. <i>arXiv preprint arXiv:2404.11584</i> .	
723		
724		
725		
726	Rui Miao, Yixin Liu, Yili Wang, Xu Shen, Yue Tan, Yiwei Dai, Shirui Pan, and Xin Wang. 2025. Blindguard: Safeguarding llm-based multi-agent systems under unknown attacks. <i>arXiv preprint arXiv:2508.08127</i> .	
727		
728		
729		
730		
731	Fatemeh Nazary, Yashar Deldjoo, and Tommaso di Noia. 2025. Poison-rag: Adversarial data poisoning attacks on retrieval-augmented generation in recommender systems. In <i>Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part IV</i> , page 239–251, Berlin, Heidelberg. Springer-Verlag.	
732		
733		
734		
735		
736		
737		
738		
739	Zepeng Ning and Lihua Xie. 2024. A survey on multi-agent reinforcement learning and its application. <i>Journal of Automation and Intelligence</i> , 3(2):73–91.	
740		
741		
742	Mengjia Niu, Hamed Haddadi, and Guansong Pang. 2025. Robust hallucination detection in llms via adaptive token selection. <i>arXiv preprint arXiv:2504.07863</i> .	
743		
744		
745		
746	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 <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> , volume 39, pages 12443–12451.	
747		
748		
749		
750		
751		
752	Junjun Pan, Yixin Liu, Yizhen Zheng, and Shirui Pan. 2023. Prem: A simple yet effective approach for node-level graph anomaly detection. In <i>2023 IEEE International Conference on Data Mining (ICDM)</i> , pages 1253–1258. IEEE.	
753		
754		
755		
756		
757	Junjun Pan, Yu Zheng, Yue Tan, and Yixin Liu. 2025b. A survey of generalization of graph anomaly detection: From transfer learning to foundation models. <i>arXiv preprint arXiv:2509.06609</i> .	
758		
759		
760		
761	Hezhe Qiao and Guansong Pang. 2023. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. <i>Advances in Neural Information Processing Systems</i> , 36:49490–49512.	
762		
763		
764		
	Hezhe Qiao, Hanghang Tong, Bo An, Irwin King, Charu Aggarwal, and Guansong Pang. 2025. Deep graph anomaly detection: A survey and new perspectives. <i>IEEE Transactions on Knowledge and Data Engineering</i> .	770
		771
		772
	Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. <i>arXiv preprint arXiv:1908.10084</i> .	773
		774
		775
		776
		777
		778
		779
		780
		781
	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In <i>Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)</i> , pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.	782
		783
		784
		785
	Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. 2023. Evil geniuses: Delving into the safety of llm-based agents. <i>arXiv preprint arXiv:2311.11855</i> .	
	Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. <i>Frontiers of Computer Science</i> , 18(6):186345.	786
		787
		788
		789
		790
	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. In <i>Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 7261–7276, Vienna, Austria. Association for Computational Linguistics.	791
		792
		793
		794
		795
		796
		797
		798
		799
	Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. <i>arXiv preprint arXiv:2502.12110</i> .	800
		801
		802
		803
	Bingyu Yan, Ziyi Zhou, Xiaoming Zhang, Chaozhuo Li, Ruilin Zeng, Yirui Qi, Tianbo Wang, and Litian Zhang. 2025. Attack the messages, not the agents: A multi-round adaptive stealthy tampering framework for llm-mas. <i>arXiv preprint arXiv:2508.03125</i> .	804
		805
		806
		807
		808
	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. <i>arXiv preprint arXiv:2505.09388</i> .	809
		810
		811
		812
		813
	Miao Yu, Shilong Wang, Guibin Zhang, Junyuan Mao, Chenlong Yin, Qijiong Liu, Qingsong Wen, Kun Wang, and Yang Wang. 2024a. Netsafe: Exploring the topological safety of multi-agent networks. <i>arXiv preprint arXiv:2410.15686</i> .	814
		815
		816
		817
		818

- 819 Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng,  
820 Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Su-  
821 chow, Zhenyu Cui, Rong Liu, and 1 others. 2024b.  
822 Fincon: A synthesized llm multi-agent system with  
823 conceptual verbal reinforcement for enhanced finan-  
824 cial decision making. *Advances in Neural Informa-  
825 tion Processing Systems*, 37:137010–137045.
- 826 Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel  
827 Kang. 2024. [InjecAgent: Benchmarking indirect  
828 prompt injections in tool-integrated large language  
829 model agents](#). In *Findings of the Association for  
830 Computational Linguistics: ACL 2024*, pages 10471–  
831 10506, Bangkok, Thailand. Association for Compu-  
832 tational Linguistics.
- 833 Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2024.  
834 A survey of multi-agent deep reinforcement learning  
835 with communication. *Autonomous Agents and Multi-  
836 Agent Systems*, 38(1):4.

## A Related Work

### A.1 Safeguarding Multi-agent System

Despite the rapid advancement of LLM-based MAS, recent studies have revealed new security vulnerabilities, including poisoning memory (Chen et al., 2024), tool injection (Zhan et al., 2024), and communication vulnerabilities (Yan et al., 2025). To address these risks, NetSafe (Yu et al., 2024a) pioneers the study of topological safety in MAS by investigating agent hallucinations and aggregation safety phenomena. AgentSafe (Mao et al., 2025) further examined the influence of malicious information on memory subsystems and introduced the concepts of system layering and isolation in LLM-based MAS. However, its reliance on redesigned MAS topologies limits its flexibility, making it unsuitable for legacy MAS with pre-defined MAS topologies. To address this, ARGUS (Li et al., 2025) investigates the flow of misinformation in MAS communication and proposes a goal-aware reasoning defense that leverages a corrective agent to correct information without requiring additional training. However, employing additional LLM-based agents as defenders reduces efficiency and expands the attack surface, as these agents can also be attacked.

To overcome these limitations, recent works leverage graph neural networks (GNNs) to operate on agent communication graphs directly, offering an efficient and effective alternative solution for MAS defense (Wang et al., 2025; He et al., 2025). G-Safeguard (Wang et al., 2025) pioneers this field by introducing a detect-then-remediate framework, in which a GNN is trained with annotations to identify malicious agents, who are then excluded from the dialogue as defense. Later, A-Trust (He et al., 2025) introduces attention-based trust metrics to evaluate violations across six fundamental trust dimensions. While these advances significantly improve MAS trustworthiness, they require supervised training or prior attack knowledge, which may not be available in real-world MAS applications. Recently, BlindGuard (Miao et al., 2025) proposed a GNN-based unsupervised MAS defense framework that leverages multi-level contextual information and contrastive learning to defend against unknown threats.

Despite the accomplishments of GNN-based defenders, they capture only coarse-grained semantics of agents' outputs when building attributed graphs from dialog, potentially overlooking mali-

cious cues, such as privacy breaches or result manipulations, that may be hidden at the fine-grained token level.

### A.2 Unsupervised Graph Anomaly Detection

Graph Anomaly Detection (GAD) aims to identify rare or unusual patterns that significantly deviate from the majority in graph data (Qiao et al., 2025; Pan et al., 2025b). Due to the scarcity of real-world anomalies, many unsupervised GAD methods have been developed, making them well-suited for addressing challenges in MAS defense.

Existing unsupervised GAD methods can be broadly categorized into three paradigms. DOMINANT (Ding et al., 2019) pioneers the **reconstruction-based paradigm** by utilizing a graph autoencoder-centric framework. With the assumption that the reconstruction process acts as a low-pass filter that removes anomalous patterns, the distance between the reconstructed graph and the original graph can serve as a reliable metric for estimating anomaly scores. Follow-up works have refined the reconstruction-based framework to address its limitations. For example, Ada-GAD (He et al., 2024) improves the training of the autoencoder by trimming heterophily edges, thereby overcoming anomaly overfitting and the homophily trap issues. **Contrastive learning-based paradigm** instead, train the anomaly detector with the supervision of constructed negative samples that simulate abnormal patterns. For instance, CoLA (Liu et al., 2021) generates negatives by swapping the context subgraphs of normal nodes. Subsequent works enhance this framework through multi-level contrastive learning (Jin et al., 2021) or improved training efficiency (Pan et al., 2023). Recently, the **affinity-based paradigm** has achieved strong performance by using local affinity metrics as anomaly measures, capturing the inherent heterophilic nature of anomalies. For example, TAM (Qiao and Pang, 2023) defines affinity as the distance between node attributes and leverages it to guide graph topology pruning, mitigating the camouflage effect of anomalies. HUGE (Pan et al., 2025a) proposes a theory-grounded affinity measure and uses it as pseudo-labels to guide the training of GAD models with a ranking-based loss, achieving effective and robust anomaly detection.

While unsupervised node-level GAD methods are generally applicable to MAS defense, they typically assume that a consistent and universal pattern exists for the normal class, which prevents them

---

**Algorithm 1** XG-Guard: Training Phase

---

**Input:** MAS graphs  $\{\mathcal{G}_1, \dots, \mathcal{G}_N\}$ , training epochs  $E$ , batch size  $B$ , trade-off parameter  $\alpha$ , learning rate  $\text{lr}$ .

**Output:** Trained XG-Guard.

- 1: Randomly initialize parameters of the bi-level encoder.
  - 2: **for**  $epoch = 1, \dots, E$  **do**
  - 3:  $\mathcal{B} \leftarrow$  (randomly split input MAS graphs into batches of size  $B$ )
  - 4: **for** batch  $b = \{\mathcal{G}_1, \dots, \mathcal{G}_B\} \in \mathcal{B}$  **do**
  - 5: Compute sentence- and token-level representation of the agents' output via Eq. (1) and (2).
  - 6: Compute graph embeddings  $\mathbf{H}^s$  and  $\mathbf{H}^t$  via Eq. (3) and (6).
  - 7: Obtain theme prototypes  $\{\mathbf{p}_1^s, \dots, \mathbf{p}_B^s\}$  and  $\{\mathbf{p}_1^t, \dots, \mathbf{p}_B^t\}$  via Eq. (7).
  - 8: Compute positive and negative sets  $\{\mathbf{s}_1^{\text{pos}}, \dots, \mathbf{s}_B^{\text{pos}}\}$  and  $\{\mathbf{s}_1^{\text{neg}}, \dots, \mathbf{s}_B^{\text{neg}}\}$  via Eq. (12) and (13).
  - 9:  $\mathcal{L} = -\sum_{k=1}^B \log(\mathbf{s}_k^{\text{pos}}) + \alpha \log(1 - \mathbf{s}_k^{\text{neg}})$
  - 10: Back-propagate  $\mathcal{L}$  to update the parameters of XG-Guard with learning rate  $\text{lr}$ .
  - 11: **end for**
  - 12: **end for**
  - 13: **return** trained XG-Guard model
- 

939 from adapting to the diverse and context-dependent  
940 normal behaviors exhibited in MAS dialogues. Fur-  
941 thermore, they produce only black-box anomaly  
942 scores without interpretability, limiting their robust-  
943 ness and practical utility in MAS defense, thereby  
944 motivating our study.

## 945 B Overall Algorithms

946 The procedure of training and inference XG-Guard  
947 is summarized in Algorithm 1 and 2 respectively.

## 948 C Complexity Analysis

949 We discuss the time complexity of each compo-  
950 nent in XG-Guard. Let  $L$  denote the average to-  
951 ken length of an agent's output and  $N$ ,  $M$  de-  
952 note the number of nodes and edges in MAS  
953 communication graphs, respectively. For the bi-  
954 level agent encoder, obtaining sentence- and token-  
955 level embeddings through SentenceBERT (Reimers  
956 and Gurevych, 2019) requires  $\mathcal{O}(NL^2)$  operations  
957 due to the self-attention. The subsequent GNN

---

**Algorithm 2** XG-Guard: Inference Phase

---

**Input:** Trained XG-Guard, test MAS graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ .

**Output:** Agent-level anomaly scores  $\mathbf{s}_{\mathcal{G}}$  and token-level explanation scores.

- 1: Compute sentence- and token-level representa-  
tion of the agents' output via Eq. (1) and (2).
  - 2: Compute graph embeddings  $\mathbf{H}^s$  and  $\mathbf{H}^t$  via Eq.  
(3) and (6).
  - 3: Obtain theme prototypes  $\mathbf{p}^s$  and  $\mathbf{p}^t$  via Eq. (7).
  - 4: Compute sentence-level and token-level  
anomaly scores  $\mathbf{s}^s, \mathbf{s}^t$  via Eq. (8)
  - 5: Compute anomaly scores  $\mathbf{s}$  via Eq. (11).
  - 6: Compute token-level explanation score  $\{s_{i,j}^{\text{exp}}\}$   
by  $\text{Cov}(\hat{\mathbf{s}}^s, \hat{\mathbf{s}}^t) \cdot \text{dist}(\mathbf{h}_{i,j}^t, \mathbf{p}^t)$ .
  - 7: **return**  $\mathbf{s}, \{s_{i,j}^{\text{exp}}\}$
- 

costs  $\mathcal{O}(M)$  to perform message passing. For the  
anomaly detector, computing the theme prototype  
has a complexity of  $\mathcal{O}(N)$ , while computing the  
anomaly score for sentence and token levels re-  
quires  $\mathcal{O}(NL)$ . To summarize, the total time com-  
plexity is  $\mathcal{O}(NL^2 + M)$ , demonstrating that XG-  
Guard is efficient and scalable.

## D Detailed Implementation

By default, we employ the Adam opti-  
mizer (Kingma, 2014) with 20 training epochs and  
an L2 regularization weight decay of  $2 \times 10^{-4}$ .  
For MA-CSQA, the learning rate is set to  $1 \times 10^{-5}$ ,  
while for all other datasets it is  $1 \times 10^{-4}$ . The  
contrastive learning trade-off parameter  $\alpha$  is set to  
 $5 \times 10^{-5}$  for PI-GSM8K and MA-CSQA,  $1 \times 10^{-5}$   
for PI-CSQA and MA-PosionRAG, and  $1 \times 10^{-4}$   
for the remaining datasets.

## E Full Results of Ablation Study

The full ablation results are reported in Table 3. We  
observe that XG-Guard consistently outperforms  
both variants, which is consistent with the analysis  
presented in the main text.

Topology	Method	PI-CSQA	PI-MMLU	PI-GSM8K	TA-InjecAgent	MA-PoisonRAG	MA-CSQA
Chain	XG-Guard	<b>87.11</b>	<b>95.11</b>	<b>97.78</b>	<b>99.56</b>	<b>99.56</b>	<b>90.67</b>
	-Fusion	80.18	67.02	59.73	48.18	96.89	81.60
	-Token	80.44	78.04	87.56	90.67	94.67	87.73
Tree	XG-Guard	<b>89.78</b>	<b>92.00</b>	<b>97.33</b>	<b>99.56</b>	<b>99.11</b>	<b>92.89</b>
	-Fusion	78.13	67.11	61.07	48.27	96.00	78.13
	-Token	80.44	77.96	87.56	90.67	94.67	87.91
Star	XG-Guard	<b>91.11</b>	<b>92.89</b>	<b>97.33</b>	<b>99.11</b>	<b>98.67</b>	<b>96.00</b>
	-Fusion	81.33	73.24	62.04	47.29	96.89	85.78
	-Token	80.44	78.13	87.56	90.67	94.58	87.73
Random	XG-Guard	<b>90.67</b>	<b>92.89</b>	<b>98.67</b>	<b>98.67</b>	<b>99.56</b>	<b>95.56</b>
	-Fusion	79.64	67.56	61.69	47.91	96.09	83.47
	-Token	80.36	77.78	87.56	90.67	94.76	87.73

Table 3: Ablation study of key designs in XG-Guard.