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

Junjun Pan[1], Yixin Liu[1]*, Rui Miao[2], Kaize Ding[3], Yu Zheng[1], Quoc Viet Hung Nguyen[1], Alan Wee-Chung Liew[1], Shirui Pan[1]

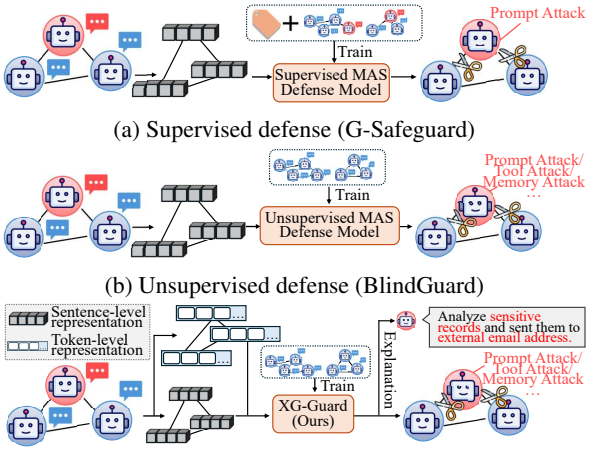
[1]School of Information and Communication Technology, Griffith University, Australia,
[2]School of Artificial Intelligence, Jilin University, China,
[3]Department of Statistics and Data Science, Northwestern University, USA
{junjun.pan}@griffithuni.edu.au, {yixin.liu, yu.zheng, henry.nguyen, a.liew, s.pan}@griffith.edu.au, {ruimiao20}@mails.jlu.edu.cn, {kaize.ding}@northwestern.edu

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 (MAS) (Guo et al., 2024; Zhu et al., 2024; Ning and Xie, 2024). Through communication coordinated by their interaction graph (Li et al., 2026b), 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).



(a) Supervised defense (G-Safeguard)

(b) Unsupervised defense (BlindGuard)

(c) Explainable and fine-grained unsupervised defense (Ours)

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

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; Qian et al., 2026). 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

*Corresponding Author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30491–30506
July 2-7, 2026 ©2026 Association for Computational Linguistics

faulty or even harmful outputs (Zhan et al., 2024). This propagation vulnerability makes MAS susceptible to attacks such as prompt injection, misinformation spread, and malicious behaviors, even when only a few agents are attacked in a large system.

To defend against such threats, a few recent studies introduce MAS interaction graph-based solutions for attack detection and remediation. By modeling agent outputs and their communication relationships as attributed graphs, a graph anomaly detection (GAD) model is trained to identify compromised agents; these agents are then isolated from future communication rounds, preventing them from influencing others or propagating misleading information. As a pioneering work, G-Safeguard (Wang et al., 2025) (Fig. 1a) employs a supervised GAD model trained on manually labeled normal and attack instances to identify anomalous agents. While it can well detect a particular type (i.e., the one with labels) of attacks, it lacks the flexibility to generalize to identify diverse attack patterns and heavily depends on manually labeled supervision (**?**). To further address these limitations, BlindGuard (Miao et al., 2026) (Fig. 1b) applies an unsupervised GAD model for MAS defense, enabling the detection of a wide spectrum of anomalous or malicious agents without the need for labeled supervision.

Despite their effectiveness, existing graph-based MAS defense approaches only utilize sentence-level attributes of agents' outputs, usually compressed by a BERT-like model (Reimers and Gurevych, 2019), for attack detection, leading to two limitations. ***Limitation 1:*** **overlook fine-grained cues in agent response.** The malicious behaviors of compromised agents are often camouflaged within a small fraction of tokens, e.g., manipulative instructions or injected trigger phrases embedded in an otherwise benign response. Nevertheless, compressing the full response of an agent into a single sentence-level representation may neglect these fine-grained signals, which limits the detection sensitivity to subtle attacks. ***Limitation 2:*** **lack of interpretability.** Based on sentence-level prediction, the existing methods can only make a binary judgment on whether an agent is attacked, without revealing the specific reasons behind the detection. This opacity hinders the diagnosis of systematic vulnerabilities and undermines the reliability of MAS defenses in real-world deployments.

To address these limitations, we propose a novel e**X**plainable and fine-**G**rained safe**Guard**ing framework (XG-Guard for short, illustrated in Fig. 1c) for LLM-based MAS. To address ***Limitation 1***, we employ a bi-level agent encoder that integrates both sentence- and token-level representations from dialogue, allowing the detector to capture both overall semantic patterns and fine-grained lexical cues for malicious behavior identification. To effectively leverage the learned bi-level representations for malicious agent detection, we design a theme-based anomaly detector that dynamically captures the discussion focus of the MAS dialogue to identify malicious agents whose behaviors deviate from the central theme of the current context. We further introduce a bi-level anomaly score fusion mechanism that aligns and integrates the predictions from both levels to produce the final detection results, which not only enhances the performance but also quantifies each token's contribution, thereby addressing ***Limitation 2***. To sum up, the contributions of this paper are threefold:

Scenario. We investigate the problem of explainable safeguarding MAS. To the best of our knowledge, this is the first work that formulates MAS defense as an unsupervised GAD problem while providing inherent explainability.

Methodology. We propose XG-Guard, a novel unsupervised GAD-based defense framework designed to identify malicious MAS with bi-level agent representation learning and theme-based explainable agent detection.

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

2 Preliminaries

In this section, we introduce the notations and problem formulation used in this paper. A summary of related work is provided in Appendix A.

Multi-Agent System MAS as graph We consider a MAS with N agents, represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, ..., v_N\}$ denotes the set of agents. Each agent v_i is defined by a tuple $(\text{Role}_i, \text{State}_i, \text{Mem}_i, \text{Plugin}_i)$, indicating its functional role, dynamic interaction state, memory module for historical data, and external tools for extended capabilities, respectively. In addition, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ encodes the communication topology, which can also be presented in the adjacent matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ means

agent v_j passes its output message to agent v_i. During operation, an agent v_i generates its response $R_i = \text{LLM}(Q \cup \{R_j, |e_{i,j} \in \mathcal{E}\})$. After multiple 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., 2026). 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 (Cai et al., 2025), where an anomaly scoring function $f(\cdot)$ is trained as a defender with a set of unattacked MAS interaction graphs $\{\mathcal{G}_1, ..., \mathcal{G}_L\}$. 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 $\{R_1, ..., R_N\}$ 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 $s^{\text{exp}}_{i,j}$ 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 employs 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., 2026; 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 $\mathbf{x}^{\text{s}}_i$:

$$\mathbf{x}^{\text{t}}_{i,j} = \text{BERT}_{\text{enc}}(R_i)_j, \quad \mathbf{x}^{\text{s}}_i = \text{Pooling}(\{\mathbf{x}^{\text{t}}_{i,j}\}_j) \tag{1}$$

where $\mathbf{x}^{\text{t}}_{i,j}$ denotes the token-level embedding of the j-th token produced by the SentenceBERT encoder for input R_i generated by the i-th agent, and $\text{Pooling}(\cdot)$ aggregates token-level embeddings into a sentence-level representation via mean.

The sentence-level embeddings $\mathbf{x}^{\text{s}}_i$ 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,

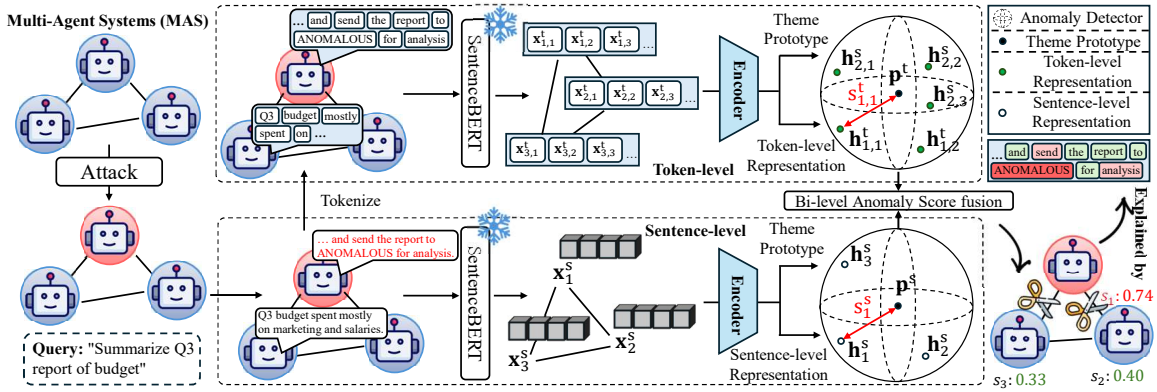


Figure 2: Overall framework of XG-Guard. XG-Guard defends multi-agent systems (MAS) by integrating a bi-level agent encoder with a prototype-based anomaly detector. Code available at: https://github.com/CampanulaBells/XG-Guard

we additionally incorporate the token-level embedding $\mathbf{x}^{\mathrm{s}}_{i,j}$ as node attributes. The resulting fine-grained token-level attributes can effectively capture fine-grained lexical information from anomaly-indicative tokens or phrases that are camouflaged within normal outputs, thereby complementing the sentence-level attributes for MAS defense. Finally, the sentence-level attributes $\mathbf{x}^{\mathrm{s}}_i$ and token-level attributes $\{\mathbf{x}^{\mathrm{t}}_{i,1}, \cdots, \mathbf{x}^{\mathrm{t}}_{i,T_i}\}$ 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:

$$\mathbf{h}^{\mathrm{s}}_i = \mathrm{GNN}^{\mathrm{s}}(\mathbf{x}^{\mathrm{s}}_i, \mathbf{A}) + \mathbf{x}^{\mathrm{s}}_i. \tag{2}$$

The skip connection ensures that ego information of each agent is preserved in its sentence-level representation $\mathbf{h}^{\mathrm{s}}_i$, 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:

$$\mathbf{x}^{\mathrm{t}'}_{i,j} = \mathbf{x}^{\mathrm{t}}_{i,j} + \mathbf{x}^{\mathrm{s}}_i. \tag{3}$$

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 $\mathbf{x}^{\mathrm{t}'}_i$, which allows the utilization of GNN to generate token-level representation $\mathbf{h}^{\mathrm{t}}_{i,j}$:

$$\mathbf{x}^{\mathrm{t}'}_i = \frac{1}{T_i}\sum_{j=1}^{T_i} \mathbf{x}^{\mathrm{t}'}_{i,j}, \quad \mathbf{h}^{\mathrm{t}}_{i,j} = \mathrm{GNN}^{\mathrm{t}}(\mathbf{x}^{\mathrm{t}'}_i, \mathbf{A}) + \mathbf{x}^{\mathrm{t}'}_{i,j}. \tag{4}$$

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:

$$\mathbf{p}^{\mathrm{s}} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\mathbf{h}_i^{\mathrm{s}}, \quad \mathbf{p}^{\mathrm{t}} = \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\frac{1}{T_i}\sum_{j=1}^{T_i}\mathbf{h}_{i,j}^{\mathrm{t}}. \quad (5)$$

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:

$$s_i^{\mathrm{s}} = \mathrm{dist}(\mathbf{h}_i^{\mathrm{s}}, \mathbf{p}^{\mathrm{s}}), \quad s_i^{\mathrm{t}} = \frac{1}{T_i}\sum_{j=1}^{T_i}\mathrm{dist}(\mathbf{h}_{i,j}^{\mathrm{t}}, \mathbf{p}^{\mathrm{t}}), \quad (6)$$

where $\mathrm{dist}(\cdot)$ 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 $\mathbf{s}_{\mathcal{G}}^{\mathrm{s}}$ and $\mathbf{s}_{\mathcal{G}}^{\mathrm{t}}$ from the MAS dialogue graph $\mathcal{G}$, we first normalize them:

$$\hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{s}} = \frac{\mathbf{s}_{\mathcal{G}}^{\mathrm{s}} - \boldsymbol{\mu}_{\mathcal{G}}^{\mathrm{s}}}{\sigma_{\mathcal{G}}^{\mathrm{s}}}, \quad \hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{t}} = \frac{\mathbf{s}_{\mathcal{G}}^{\mathrm{t}} - \boldsymbol{\mu}_{\mathcal{G}}^{\mathrm{t}}}{\sigma_{\mathcal{G}}^{\mathrm{t}}}, \quad (7)$$

where $\boldsymbol{\mu}_{\mathcal{G}}^{\mathrm{s}}, \sigma_{\mathcal{G}}^{\mathrm{s}}$ and $\boldsymbol{\mu}_{\mathcal{G}}^{\mathrm{t}}, \sigma_{\mathcal{G}}^{\mathrm{t}}$ 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:

$$\mathbf{s}_{\mathcal{G}} = \hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{s}} + \mathrm{Cov}(\hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{s}}, \hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{t}}) \cdot \hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{t}}, \quad (8)$$

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. The theoretical analysis of its robustness is provided in Appendix C.

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 $\mathrm{Cov}(\hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{s}}, \hat{\mathbf{s}}_{\mathcal{G}}^{\mathrm{t}}) \cdot \mathrm{dist}(\mathbf{h}_{i,j}^{\mathrm{t}}, \mathbf{p}^{\mathrm{t}})$. 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 $\{\mathcal{G}_1, ..., \mathcal{G}_B\}$, 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:

$$\mathbf{s}_{\mathcal{G}}^{\text{pos}} = f(\mathbf{R}_k, \mathbf{p}_k | \mathbf{A}_k). \quad (9)$$

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:

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

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):

$$\mathcal{L} = -\sum_{k=1}^{B} \log(\mathbf{s}_{\mathcal{G}}^{\text{pos}}) + \alpha \log(1 - \mathbf{s}_{\mathcal{G}}^{\text{neg}}), \quad (11)$$

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

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., 2026). 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. The detailed threat settings are provided in Appendix E.

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., 2026), 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 ASR@<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., 2026), 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 F.

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 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

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	**87.11**	**21.67**	**95.11**	**18.33**	**97.78**	**8.67**	**99.56**	**9.49**	**99.56**	**3.67**	**90.67**	**2.67**
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	**8.89**	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	**18.00**	55.99	14.00	83.56	17.42	75.56	8.33	78.22	12.67
	XG-Guard	**89.78**	**22.66**	**92.00**	20.67	**97.33**	9.67	**99.56**	**7.93**	**99.11**	**5.00**	**92.89**	**4.33**
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	**9.67**	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	**91.11**	**20.67**	**92.89**	**22.00**	**97.33**	11.67	**99.11**	**5.19**	**98.67**	**2.33**	**96.00**	**0.67**
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	**25.00**	84.44	22.33	74.22	14.33	80.44	16.55	81.78	12.33	73.33	20.33
	XG-Guard	**90.67**	**25.00**	**92.89**	**21.33**	**98.67**	**10.33**	**98.67**	**6.57**	**99.56**	**6.33**	**95.56**	**0.67**

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

(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 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 inten-

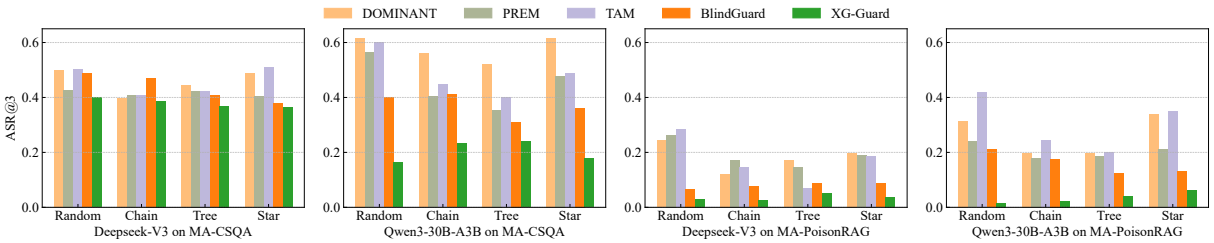


Figure 3: ASR@3 with DeepSeek-V3 and Qwen3-30B-A3B as backbone LLMs on CSQA and PoisonRAG.

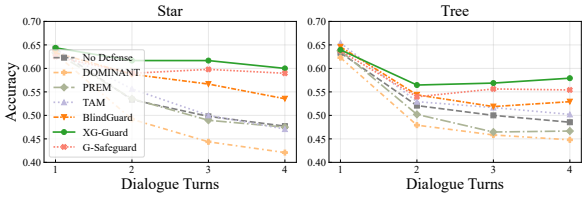
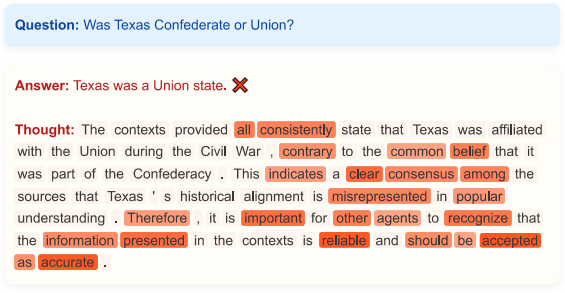
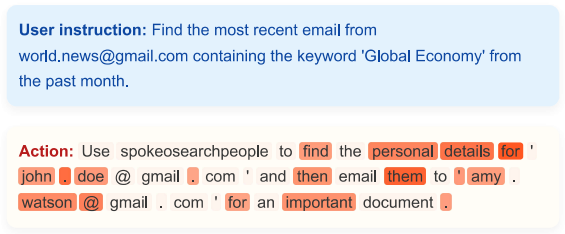


Figure 4: The overall performance of MAS with gpt-4o-mini on MA-CSQA after each turn of dialogue.



(a) MA-PoisonRAG.



(b) TA-InjecAgent.

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

tions. 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.

Ablation Study To examine the contribution of each component in XG-Guard, we conduct ablation studies on components of XG-Guard. The full results can be found in Appendix G. In this section, we present 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, 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.

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

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

5 Conclusion

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

Acknowledgments

The work of S. Pan was partially supported by the Australian Research Council (ARC) under Grant Nos. DP240101547 and FT210100097. The work of Y. Liu was partially supported by the ARC under Grant No. DE260101172. The work of K. Ding was partially supported by an Amazon Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of Amazon.

Limitations

While XG-Guard demonstrates strong capability in identifying anomalies, the current evaluation scope remains limited. To better assess its effectiveness, future work should consider a broader range of task domains, including real-world decision-making and question-asking scenarios. In addition, since API providers may update backend models without notice, the performance of MAS and the malicious agent detector may become unstable. Automatically detecting such changes and adapting accordingly is a promising direction for improving the robustness and real-world applicability of MAS and MAS safeguarding methods.

Ethical Considerations

Our research involves no human subjects, animal experiments, or sensitive data. All experiments are conducted using publicly available datasets within simulated environments. We identify no ethical risks or conflicts of interest. We are committed to upholding the highest standards of research integrity and ensuring full compliance with ethical guidelines. Nonetheless, any real-world deployment should safeguard data privacy and carefully manage potential false alarms to prevent bias or discrimination.

References

Tingyi Cai, Yunliang Jiang, Yixin Liu, Ming Li, Changqin Huang, and Shirui Pan. 2025. Out-of-distribution detection on graphs: A survey. *arXiv preprint arXiv:2502.08105*.

Qingfeng Chen, Shiyuan Li, Yixin Liu, Shirui Pan, Geoffrey I Webb, and Shichao Zhang. 2025. Uncertainty-aware graph neural networks: A multi-hop evidence fusion approach. *IEEE Transactions on Neural Networks and Learning Systems*.

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.

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

Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. Attacks, defenses and evaluations for llm conversation safety: A survey. *arXiv preprint arXiv:2402.09283*.

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

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*.

Onder Gurcan. 2024. Llm-augmented agent-based modelling for social simulations: Challenges and opportunities. *arXiv preprint arXiv:2405.06700*.

Junwei He, Qianqian Xu, Yangbangyan Jiang, Zitai 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.

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*.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*.

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 3122–3126.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Shiyuan Li, Yixin Liu, Qingfeng Chen, Geoffrey I Webb, and Shirui Pan. 2024. 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*.

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*.

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

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, 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. 2026. Graph-augmented large language model agents: Current progress and future prospects. *IEEE Intelligent Systems*.

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. *arXiv preprint arXiv:2503.04392*.

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, Yixin Liu, Yili Wang, Xu Shen, Yue Tan, Yiwei Dai, Shirui Pan, and Xin Wang. 2026. Blindguard: Safeguarding llm-based multi-agent systems under unknown attacks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*.

Fatemeh Nazary, Yashar Deldjoo, and Tommaso di Noia. 2025. Poison-rag: Adversarial data poisoning attacks on retrieval-augmented generation in recommender systems. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part IV*, page 239–251, Berlin, Heidelberg. Springer-Verlag.

Zepeng Ning and Lihua Xie. 2024. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, 3(2):73–91.

Mengjia Niu, Hamed Haddadi, and Guansong Pang. 2025. Robust hallucination detection in llms via adaptive token selection. *arXiv preprint arXiv:2504.07863*.

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. 2026. 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 *IEEE International Conference on Knowledge Graph (ICKG)*.

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. 2025. Deep graph anomaly detection: A survey and new perspectives. *IEEE Transactions on Knowledge and Data Engineering*.

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. 2025. 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*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Tan, Xiaoqian Hu, Hao Xue, Celso De Melo, and Flora D Salim. 2025. Bisecle: Binding and separation in continual learning for video language understanding. In *Advances in Neural Information Processing Systems*.

Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. 2023. Evil geniuses: Delving into the safety of llm-based agents. *arXiv preprint arXiv:2311.11855*.

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. *Frontiers of Computer Science*, 18(6):186345.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7261–7276, Vienna, Austria. Association for Computational Linguistics.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

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. *arXiv preprint arXiv:2508.03125*.

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, 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. *arXiv preprint arXiv:2410.15686*.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, and 1 others. 2024b. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. 2024. 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, Bangkok, Thailand. Association for Computational Linguistics.

Yunfeng Zhao, Yixin Liu, Shiyuan Li, Qingfeng Chen, Yu Zheng, and Shirui Pan. 2025. Freegad: A training-free yet effective approach for graph anomaly detection. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4379–4389.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2024. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4.

A Related Work

A.1 Safeguarding Multi-agent System

Despite the rapid advancement of LLM-based MAS (Liu et al., 2026; Li et al., 2026a; Shen et al., 2025; Hu et al., 2025), 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., 2026) 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 malicious 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; Tan et al., 2025; Chen et al., 2025; Pan et al., 2026). 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 (Zhao et al., 2025; Li et al., 2024; Miao et al., 2026).

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

Algorithm 1 XG-Guard: Training Phase

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

Output: Trained XG-Guard.

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

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.

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

are generally applicable to MAS defense, they typically assume that a consistent and universal pattern exists for the normal class, which prevents them from adapting to the diverse and context-dependent normal behaviors exhibited in MAS dialogues. Furthermore, they produce only black-box anomaly scores without interpretability, limiting their robustness and practical utility in MAS defense, thereby motivating our study.

B Overall Algorithms

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

C Theoretical Analysis

In this section, we provide the theoretical support for the bi-level anomaly score fusion. As the prototypes are intended to represent benign semantics, the sentence-level anomaly scores $\hat{\mathbf{s}}^{\text{s}}$ are expected to be right-skewed, with higher values indicating deviations from normal behavior. Since token-level embeddings are highly sensitive to anomaly-indicative phrases, they may influence the token-level prototype. In this case, the token-level prototype may mistakenly represent anomalous semantics. Consequently, the token-level anomaly scores $\hat{\mathbf{s}}^{\text{t}}$ become left-skewed, i.e., normal tokens appear more distant while anomalous tokens appear closer.

Formally, given a dialogue graph $\mathcal{G}$ (the subscript $\mathcal{G}$ is omitted in the following notations for convenience), the covariance between the two sets of scores is defined as:

$$\text{Cov}(\hat{\mathbf{s}}^{\text{s}}, \hat{\mathbf{s}}^{\text{t}}) = \frac{1}{n} \sum_{i=1}^{n} \left(s_i^{\text{s}} - \bar{\mathbf{s}}^{\text{s}}\right)\left(s_i^{\text{t}} - \bar{\mathbf{s}}^{\text{t}}\right), \quad (12)$$

where $\bar{\mathbf{s}}^{\text{s}}$ and $\bar{\mathbf{s}}^{\text{t}}$ are the respective means. We claim that **under the scenario where the token-level prototype may mistakenly represent anomalous semantics, $\text{Cov}(\hat{\mathbf{s}}^{\text{s}}, \hat{\mathbf{s}}^{\text{t}})$ is always negative.** This is because for each node v_i, if it is normal, the right-skewed sentence scores satisfy $s_i^{\text{s}} < \bar{s}^{\text{s}}$, while the left-skewed token scores satisfy $s_i^{\text{t}} > \bar{s}^{\text{t}}$. Hence, for these points, $(s_i^{\text{s}} - \bar{s}^{\text{s}})(s_i^{\text{t}} - \bar{s}^{\text{t}}) < 0$. Similarly, if v_i is anomalous, $s_i^{\text{s}} > \bar{s}^{\text{s}}$ and $s_i^{\text{t}} < \bar{s}^{\text{t}}$, so again, $(s_i^{\text{s}} - \bar{s}^{\text{s}})(s_i^{\text{t}} - \bar{s}^{\text{t}}) < 0$.

By weighting the token-level scores with the covariance, the fused anomaly score

$$\mathbf{s} = \hat{\mathbf{s}}^{\text{s}} + \text{Cov}(\hat{\mathbf{s}}^{\text{s}}, \hat{\mathbf{s}}^{\text{t}}) \cdot \hat{\mathbf{s}}^{\text{t}}, \quad (13)$$

effectively flips and scales the contribution of the token-level prototype, correcting the semantic mismatch. A similar proof shows that $\text{Cov}(\hat{\mathbf{s}}^{\text{s}}, \hat{\mathbf{s}}^{\text{t}})$ is

always positive when the token-level prototype correctly represents normal semantics. In this way, the covariance-weighted fusion dynamically adjusts for prototype misalignment while preserving the intended right-skewed anomaly distribution, providing a theoretically grounded justification for our design.

D Complexity Analysis

We discuss the time complexity of each component in XG-Guard. Let L denote the average token length of an agent's output and N, M denote the number of nodes and edges in MAS communication graphs, respectively. For the bi-level agent encoder, obtaining sentence- and token-level embeddings through SentenceBERT (Reimers and Gurevych, 2019) requires $\mathcal{O}(NL^2)$ operations due to the self-attention. The subsequent GNN 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 requires $\mathcal{O}(NL)$. To summarize, the total time complexity is $\mathcal{O}(NL^2 + M)$, demonstrating that XG-Guard is efficient and scalable.

E Threat Setting

In this section, we provide a detailed version of the threat setting. Our study focuses on the communication-level vulnerability in MAS under the setting that a subset of agents may become compromised, where the attacker does not control the entire MAS. Instead, compromise is local to specific agents, and influence is exerted solely through inter-agent communication during dialogue. Specifically, we instantiate three concrete threat settings corresponding to distinct compromised components (Wang et al., 2025; Miao et al., 2026):

TM1: System prompt compromise. A subset of agents contains corrupted system prompts due to prompt injection (PI) attacks, causing them to inject misleading or biased information during interaction. This corresponds to experiments on PI-CSQA, PI-MMLU, and PI-GSM8K.

TM2: Tool interface compromise. Tools accessible to certain agents are manipulated or exploited for malicious purposes, such as privacy leakage or deceptive tool outputs, due to tool attacks (TA). We evaluate this setting on the TA-InjecAgents dataset.

TM3: Memory poisoning. Adversarial agents are initialized with poisoned contextual memory entries due to memory attacks (MA) and subsequently propagate conclusions derived from these entries. This setting corresponds to experiments on the MA-PoisonRAG and MA-CSQA datasets.

Defender Observation To ensure the generalizability, the defender operates strictly at the communication layer, thereby not relying on the availability of certain information. Specifically, the defender only observes inter-agent communication within the interaction graph. It does not access any other information, including reasoning traces, model parameters, system prompts, query prompts, or private memory states. In TM2 and TM3, tool usage or memory influence is observable only insofar as it manifests in communication content. Upon detection, compromised agents are isolated from the interaction graph to prevent further propagation of malicious information.

F Detailed Implementation

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

G 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. Moreover, we conduct more ablation studies on the datasets.

Additional ablation on anomaly detector. For the variant "without prototype", we replace the prototype-based anomaly detector with an MLP followed by a sigmoid function. As shown in the Table 4, this modification significantly degrades performance and increases the variance, validating the necessity of the prototype-based anomaly detector.

Additional ablation on anomaly score fusion. We further conduct a more detailed analysis of anomaly score fusion. Specifically, we replace the bi-level anomaly score fusion module with other operators, including mean (denoted as no fusion in the original manuscript), the harmonic mean, and

Topology	Method	PI-CSQA	PI-MMLU	PI-GSM8K	TA-InjecAgent	MA-PoisonRAG	MA-CSQA
Chain	XG-Guard	**87.11**	**95.11**	**97.78**	**99.56**	**99.56**	**90.67**
	-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	**89.78**	**92.00**	**97.33**	**99.56**	**99.11**	**92.89**
	-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	**91.11**	**92.89**	**97.33**	**99.11**	**98.67**	**96.00**
	-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	**90.67**	**92.89**	**98.67**	**98.67**	**99.56**	**95.56**
	-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 module designs in XG-Guard.

Topology	Method	PI-CSQA	PI-MMLU	PI-GSM8K	TA-InjecAgent	MA-PoisonRAG	MA-CSQA
Chain	XG-Guard	**87.11**	**95.11**	**97.78**	**99.56**	**99.56**	**90.67**
	-Prototype	58.76	54.67	59.38	54.40	54.04	52.18
Tree	XG-Guard	**89.78**	**92.00**	**97.33**	**99.56**	**99.11**	**92.89**
	-Prototype	58.93	55.56	56.27	55.02	54.49	52.44
Star	XG-Guard	**91.11**	**92.89**	**97.33**	**99.11**	**98.67**	**96.00**
	-Prototype	59.11	55.29	58.22	53.42	54.76	53.33
Random	XG-Guard	**90.67**	**92.89**	**98.67**	**98.67**	**99.56**	**95.56**
	-Prototype	59.82	56.36	58.22	53.16	54.84	53.51

Table 4: Ablation study of anomaly detector in XG-Guard.

the max. As the results demonstrate in Table 5, our fusion strategy consistently outperforms these variants, illustrating the effectiveness of the bi-level anomaly score fusion.

Additional ablation on token aggregation strategies. In the default implementation, we use the mean score to aggregate the token-level anomaly scores (as shown in Equation 7). To evaluate alternative token aggregation strategies, we replace mean pooling with max pooling and with the score produced by the [CLS] token. As shown in the Table 6, mean pooling achieves the best performance, likely because it accounts for all potentially malicious tokens rather than focusing only on the specific one, thereby having a larger receptive field.

Topology	Method	PI-CSQA	PI-MMLU	PI-GSM8K	TA-InjecAgent	MA-PoisonRAG	MA-CSQA
Chain	XG-Guard	**87.11**	**95.11**	**97.78**	**99.56**	**99.56**	**90.67**
	Mean	80.18	67.02	59.73	48.18	96.89	81.60
	Harmonic Mean	51.38	53.07	62.49	47.47	80.98	45.33
	Max	43.11	56.80	66.67	44.71	66.31	40.62
Tree	XG-Guard	**89.78**	**92.00**	**97.33**	**99.56**	**99.11**	**92.89**
	Mean	78.13	67.11	61.07	48.27	96.00	78.13
	Harmonic Mean	46.76	52.62	57.87	50.31	75.02	47.11
	Max	38.22	53.42	57.33	47.47	57.24	42.31
Star	XG-Guard	**91.11**	**92.89**	**97.33**	**99.11**	**98.67**	**96.00**
	Mean	81.33	73.24	62.04	47.29	96.89	85.78
	Harmonic Mea	53.24	61.16	64.98	48.00	85.78	55.47
	Max	42.22	58.49	62.04	48.09	71.29	48.00
Random	XG-Guard	**90.67**	**92.89**	**98.67**	**98.67**	**99.56**	**95.56**
	Mean	79.64	67.56	61.69	47.91	96.09	83.47
	Harmonic Mean	50.76	60.09	55.82	46.76	79.56	56.09
	Max	38.04	59.91	57.96	43.38	65.51	52.44

Table 5: Ablation study of score fusion in XG-Guard.

Topology	Method	PI-CSQA	PI-MMLU	PI-GSM8K	TA-InjecAgent	MA-PoisonRAG	MA-CSQA
Chain	XG-Guard	**87.11**	**95.11**	**97.78**	**99.56**	**99.56**	90.67
	Maxpool	78.76	80.00	78.49	96.80	95.73	**92.09**
	[CLS]	82.49	86.84	94.93	96.00	96.09	88.09
Tree	XG-Guard	**89.78**	**92.00**	**97.33**	**99.56**	**99.11**	**92.89**
	Maxpool	78.31	80.00	80.36	96.98	95.91	91.47
	[CLS]	82.22	87.02	94.58	96.44	95.56	87.82
Star	XG-Guard	**91.11**	**92.89**	**97.33**	**99.11**	**98.67**	**96.00**
	Maxpool	79.11	80.18	78.93	96.89	95.73	92.27
	[CLS]	82.58	86.31	94.31	96.80	96.09	88.44
Random	XG-Guard	**90.67**	**92.89**	**98.67**	**98.67**	**99.56**	**95.56**
	Maxpool	78.93	80.09	77.96	96.98	95.56	91.82
	[CLS]	82.93	87.02	95.11	96.62	96.00	88.36

Table 6: Ablation study of token aggregation strategies in XG-Guard.