MITIGATING SPARSITY AND HETEROGENEITY IN UNSUPERVISED FEDERATED GRAPH ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated graph anomaly detection shows promising potential in academia and industry since it solves the limitations of traditional centralized GAD. However, existing methods suffer from local sparsity, intrinsic distribution shifts, and label reliance issues, resulting in limited semantic information in local clients, strong heterogeneity across clients, and vulnerability to label quality. To address these issues, we propose an unsupervised federated graph anomaly detection (UnFedGAD) framework, which enhances local semantic information through cross-client topology suture and addresses heterogeneity via personalized bidirectional parameter updating. Specifically, UnFedGAD discovers adequate semantics via cross-client communication with privacy protection, identifying the nearest neighboring client based on prototype similarity for each client. In addition, UnFedGAD employs prototype similarities to compute adaptive weights for global consensus generation and introduces a novel client updating mechanism to strike a dynamic balance between local and global models, effectively conquering client heterogeneity. To eliminate the dependence on labels, UnFedGAD generates pseudo-labels based on local-global consistency estimation for unsupervised training. Extensive experiments on eight public datasets demonstrate the effectiveness and advancement of UnFedGAD compared with nine baselines.

1 INTRODUCTION

Graph anomaly detection (GAD) plays a vital role in protecting system robustness and reducing potential economic or security losses by identifying abnormal patterns within complex graph data, gaining substantial research potential in extensive industrial applications, such as fraud detection (Zhang & Luo, 2025; Zhang et al., 2025d), financial risk identification (Yuan et al., 2025; Zhang et al., 2025e), and intrusion detection (Pei et al., 2025). Existing supervised and unsupervised GAD methods usually follow centralized learning that assumes all datasets are shared with a single model, inevitably raising serious data privacy concerns.

Federated graph learning (FGL) aims to facilitate collaborative model training on multiple independent data owners via the interaction of model parameters. Recently, some studies have integrated FGL into the GAD, eliminating the risks of data disclosure (Kong et al., 2025; Wu et al., 2025). Although they achieve promising performance, they are subject to some limitations. First, sparsity is a primary limitation in federated settings, which is caused by the inaccessibility of cross-client data. This directly leads to inadequate interactive and semantic information within local graphs, missing discriminative behavior patterns of nodes. Second, due to the intrinsic distribution shift of federated settings, clients exhibit strong heterogeneity and non-IID properties, which pose grim challenges for generating a comprehensive global consensus and capturing generalized characteristics for each client. Third, the reliance on high-quality label information results in model vulnerability, especially in scenarios where labels are unavailable or contaminated.

To alleviate these limitations, we need to tackle threefold challenges. **(1) How to supplement redundant semantic information for each client.** To improve the limited representation learning capabilities of the local model caused by insufficient data accessibility, we aim to explore cross-client nodes with sufficient semantic information as sutured neighbors by determining a high-quality

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

client based on the prototype quality evaluation and threshold selection. Simultaneously, we need to achieve cross-dataset feature sharing without privacy leakage. **(2) How to overcome heterogeneity across clients in federated settings and generate a comprehensive consensus.** The heterogeneity makes it inherently difficult to form a comprehensive global consensus. Furthermore, directly imposing this global consensus as the updated local parameters may overlook client-specific distinct characteristics. We utilize prototypes as evaluation information of client-specific contributions to address the heterogeneity issue. In parallel, we design a weighted client update strategy that seeks to adaptively balance global consensus with local knowledge. **(3) How to perform local anomaly detection in an unsupervised manner.** Although unsupervised GAD is fully explored under centralized learning, it remains challenging in a federated setting because of the requirement for substantial normal samples. Given that samples are separated across distributed clients in federated GAD, we introduce a pseudo-labeling strategy to alleviate the reliance on labels.

Based on the above insights, we propose an unsupervised federated graph anomaly detection (UnFedGAD) framework. The core of UnFedGAD lies in supplementing semantics via cross-client topology suture and alleviating heterogeneity in federated environments in an unsupervised manner. On the one hand, UnFedGAD designs a cross-client topology suture module for enhancing local semantic propagation, which identifies the nearest neighboring client based on prototype similarity and retrieves corresponding affinity nodes as semantic supplements. On the other hand, UnFedGAD employs prototype similarity to generate adaptive weights for bidirectional parameter updating, deriving a comprehensive global consensus and achieving the global-local balance. For local training, UnFedGAD generates pseudo-labels based on local-global consistency scores, alleviating the dependence of FGL on labels. Our main contributions are as follows:

- **New research task**. We attempt to simultaneously address sparsity and heterogeneity issues in federated GAD with unsupervised paradigms.
- **Novel FGAD framework**. UnFedGAD is proposed, which introduces a topology suture strategy to enhance local semantic propagation and develops a prototype-based personalized bidirectional updating mechanism to obtain comprehensive global consensus and achieve global-local balance.
- **Promising performance**. Extensive experiments on comprehensive public datasets demonstrate the effectiveness and advancement of UnFedGAD compared with nine baselines.

2 Related Work

2.1 Graph Anomaly Detection

Traditional GAD approaches are primarily categorized into supervised and unsupervised paradigms. Supervised methods fully exploit label information to learn discriminative latent representations for anomalies. For instance, U-A2GAD (Li et al., 2025c) emphasizes spectral augmentation, leveraging label guidance to shape latent feature spaces of graph signals. HeteGAD (Huang et al., 2025) addresses relational heterophily by designing both heterophily-aware and homophily-aware aggregation mechanisms, enabling effective modeling of complex interaction structures. GE-GNN (Zhang & Luo, 2025) exploits semantic edge information and employs a gating mechanism to selectively refine anomalous node representations. Unsupervised GAD methods aim to characterize normal patterns based on high-quality normal samples and elaborate specific criteria to evaluate anomalies. These approaches can be mainly grouped into reconstruction-based methods and contrastive learning methods. Among graph anomaly measure–based approaches. Reconstruction-based methods focus on modeling normality through graph reconstruction. For example, G3AD (Bei et al., 2025) designs a multi-channel reconstruction framework, including feature- and topology-level tasks, and introduces global alignment to enforce intra-view consistency. Self-supervised GAD methods typically leverage contrastive learning to uncover abnormal patterns without relying on labeled data. MCLASt (Zhang & Luo, 2026) develops hierarchical contrastive objectives and introduces topology awareness to capture anomalies within complex graph structures. CARE (Zheng et al., 2025) integrates clustering-based contrastive learning with topology reconstruction to achieve local–global alignment, while GCLAD (Xiao et al., 2025) employs a diffusion probabilistic model to generate augmented samples and develops a multi-scale contrastive learning based on meta-paths to enhance

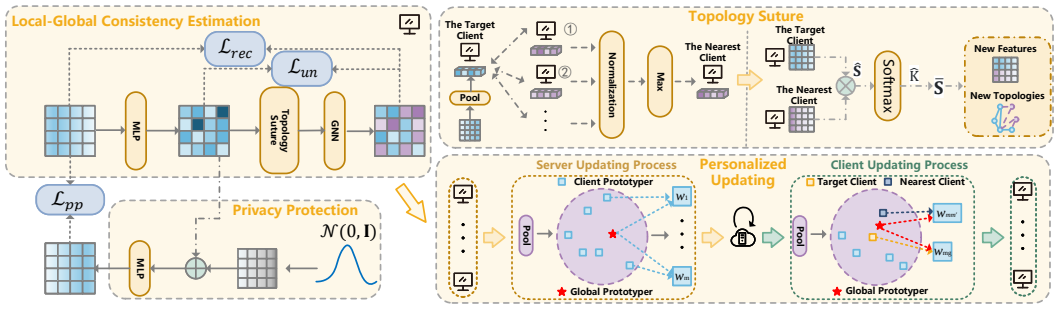

Figure 1: The framework of UnFedGAD.

topology consistency learning. Despite their effectiveness, most existing GAD methods follow a centralized learning paradigm, which entails strict access to private individual data, potentially posing significant privacy concerns.

2.2 Federated Graph Learning

FGL provides a solution for collaboratively training distributed graph models across multiple clients while preserving data privacy, which has attracted increasing attention due to its wide-ranging research and practical implications (Li et al., 2025b). The core of FGL is to design effective interaction mechanisms that enable the server to capture global consensus from distributed graph samples. Substantial efforts have been devoted to discriminating global knowledge consensus (Zhang et al., 2025a; Yue et al., 2025). For instance, FedVN (Fu et al., 2025) introduces virtual nodes as shared representations to alleviate the adverse effects of distribution shifts. FedMetro (Zhang et al., 2025c) introduces cross-attention-based federated dynamic graph learning methods to model spatiotemporal correlations. S2FGL (Tan et al., 2025) extends the FGL into the spectral domain.

Some studies introduce federated learning into the graph anomaly detection problem. FedCAD (Kong et al., 2025) exploits intra-view contrastive learning to learn the complex patterns of nodes in the graph. FedCLGN (Wu et al., 2025) generates global negative samples to address the structure sparsity and utilizes consistency scores based on contrastive learning to evaluate anomalies. However, these federated GAD methods have limited capacity to capture adequate semantics for local models, resulting in a limited ability to identify discriminative patterns of anomalies. On the one hand, they overlook the exploration of valuable cross-client semantics to alleviate sparsity problems. On the other hand, existing methods fail to address heterogeneity issues because they utilize gradient or statistical information to achieve directional updating, while overlooking the evaluation of client-specific semantic quality and ignoring the balance between global consensus and local modeling. Additionally, some existing federated graph anomaly detection studies (Cai et al., 2024b;a; Liu et al., 2025; Cai et al., 2024a) actually focus on graph-level anomaly detection, and their detailed analysis is in Section A.

3 Methodology

3.1 Preliminaries

Define a graph $\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{A}, \mathbf{y})$. $\mathcal{V} = \{v_1, \cdots, v_N\}$ is the node set, where N is the node number. $\mathbf{X} \in \mathbb{R}^{N \times F}$ represents the node feature matrix, where F is the original feature dimension. $\mathbf{A} \in \{0,1\}^{N \times N}$ represents the adjacency matrix without self-loop. $\mathbf{y} \in \{0,1\}^N = \{y_1, \cdots, y_N\}$ denotes the labels of nodes, where $y_i = 0$ denote node v_i is normal node and $y_i = 1$ otherwise. In the federated graph anomaly detection problem, we define a client set $\mathcal{C} = \{C_1, \cdots, C_M\}$, where M is the number of clients. Each client C_m holds a private subgraph $\mathcal{G}^m = (\mathcal{V}^m, \mathbf{X}^m, \mathbf{A}^m, \mathbf{y}^m)$. $\mathcal{V}^m = \{v_1, \cdots, v_{N_m}\} \subset \mathcal{V}$ is node set of C_m, where N_m is the number of nodes in client C_m. $\mathbf{X}^m \in \mathbb{R}^{N_m \times F}$, $\mathbf{A}^m \in \{0,1\}^{N_m \times N_m}$, and $\mathbf{y}^m \in \{0,1\}^{N_m} = \{y_1, \cdots, y_{N_m}\}$. For any local subgraph $\mathcal{G}^m$ of client C_m, $\mathcal{G}^m \subseteq \mathcal{G}$, indicating that the subgraph $\mathcal{G}^m$ is a subset of $\mathcal{G}$. For any subgraph $\mathcal{G}^{m_1}$, there may exist $m_2 \neq m_1$ such that $\mathcal{V}^{m_1} \cap \mathcal{V}^{m_2} \neq \varnothing$, which implies that the same node may

appear across clients. Federated GAD aims to learn an anomaly score vector $\mathbf{s}^m(m = 1, \cdots, M)$ for each client using local models to evaluate the anomaly degree for every node, where a larger value of s_i^m suggests a higher abnormality for node v_i^m.

3.2 OVERVIEW

The framework of UnFedGAD is shown in Fig. 1. The core of UnFedGAD is to explore cross-client semantic nodes to suture the local incomplete topology, capturing more affinity information about nodes. In addition, UnFedGAD generates pseudo-labels to improve local-global consistency estimation. A personalized bidirectional updating mechanism is used to comprehensively extract knowledge from all clients, which leverages prototype similarity to calculate the dynamic weights for global consensus generation and introduces client updating to achieve bidirectional updating.

3.3 CROSS-CLIENT TOPOLOGY SUTURE

In federated graph learning, the inaccessibility of cross-client data due to privacy constraints leads to severe sparsity in each client's local data. Such sparsity significantly weakens the model's ability to capture rich semantics, rendering unsupervised consistency learning over limited local features both insufficient and impractical. To address this challenge, we propose a cross-client topology suture strategy, which builds connections with cross-client affinity nodes that carry redundant semantics, thereby enhancing the interactive representations of nodes. Achieving this goal requires solving two key issues. The first is how to choose an appropriate neighboring client under the distribution drift across clients. We use prototype similarity to select the nearest client, as prototypes provide comprehensive client-level information. The second is how to identify affinity nodes inside the selected client. We employ affinity thresholds to filter highly semantic nodes and suture the local topologies accordingly.

Specifically, for client C_m, we map the node into common space using the MLP to encode the original features:

$$\mathbf{H}^m = \mathrm{MLP}(\mathbf{X}^m), \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{N_m \times d}$ is the node features with latent dimension. To achieve effective communication between different clients without exposing local private data, we design a privacy protection mechanism to upload the node features to the server. First, we add standard Gaussian noise to $\mathbf{H}^m$ based on the reparameterization trick to generate $\hat{\mathbf{H}}^m \in \mathbb{R}^{N \times d}$ without gradient clipping, and utilize an $\mathrm{MLP}_{\text{diff}}$ to regularize the encrypted representations:

$$\hat{\mathbf{H}}^m = \mathbf{H}^m + \sigma\mathbf{E}, \quad \hat{\mathbf{X}}^m = \mathrm{MLP}_{\text{diff}}(\hat{\mathbf{H}}^m), \tag{2}$$

where $\mathbf{E} \sim \mathcal{N}(0, \mathbf{I}) \in \mathbb{R}^{N_m \times d}$ and σ is a learnable parameter, and $\hat{\mathbf{X}}^m \in \mathbb{R}^{N_m \times F}$ is the reconstruction outputs. The loss function of the privacy protection mechanism in the local client can be formulated as:

$$\mathcal{L}_{\text{pp}} = \mathcal{L}_{\text{mlp}} + \mathcal{L}_{\text{IB}}. \tag{3}$$

where $\mathcal{L}_{\text{mlp}} = ||\hat{\mathbf{X}}^m - \mathbf{X}^m||_2$ is to capture the critical characteristics of the original features based on MSE regression and $\mathcal{L}_{\text{IB}} = I(\mathbf{X}^m, \hat{\mathbf{H}}^m) - 0.01 * I(\hat{\mathbf{X}}^m, \hat{\mathbf{H}}^m)$ is designed based on information bottleneck to ensure the privacy of local data, which protects privacy by minimizing $I(\mathbf{X}^m, \hat{\mathbf{H}}^m)$, and preserves enough information to make the encrypted embeddings $\hat{\mathbf{H}}^m$ representative enough by maximizing $I(\hat{\mathbf{X}}^m, \hat{\mathbf{H}}^m)$.

Through the privacy protection mechanism, clients can upload the encrypted latent representations $\hat{\mathbf{H}}^m$ to the server without privacy concerns. Based on this mechanism, we then calculate the prototype $\hat{\mathbf{c}}^m \in \mathbb{R}^d$ of the client C_m:

$$\hat{\mathbf{c}}^m = \mathrm{Mean}(\hat{\mathbf{H}}^m), \tag{4}$$

where $\mathrm{Mean}(\cdot)$ is mean function. Accordingly, traversing all clients, the prototype set of all clients $\mathcal{C} = \{\hat{\mathbf{c}}_\mathbf{1}, \cdots, \hat{\mathbf{c}}_\mathbf{M}\}$ can be generated on the server. For client C_m, we calculate the inner product between C_m and the other clients to find the nearest neighboring client, denoted as $C_{m'}$:

$$\max_{m' \in \{1, \cdots, M\}, m' \neq m} \hat{\mathbf{c}}^m (\hat{\mathbf{c}}^{m'})^\top, m = 1, \cdots, M. \tag{5}$$

Then, we construct the affinity matrix of node features between C_m and $C_{m'}$ based on the encrypted $\hat{\mathbf{H}}^m$ and obtain the sutured topology $\bar{\mathbf{S}} \in \mathbb{R}^{N_m \times N_{m'}}$ to discover high-quality semantic-related neighbors from client $C_{m'}$ by sparsification based on the affinity threshold $\hat{K}$:

$$\hat{\mathbf{S}} = \mathrm{Softmax}(\hat{\mathbf{H}}^m(\hat{\mathbf{H}}^{m'})^\top), \quad \bar{s}_{ij} = \begin{cases} 0, & \text{if} \quad \hat{s}_{ij} < \hat{K}, \\ 1, & \text{if} \quad \hat{s}_{ij} \geq \hat{K}. \end{cases} \tag{6}$$

where $\mathrm{Softmax}(\cdot)$ is normalization function and $\hat{\mathbf{S}} \in \mathbb{R}^{N_m \times N_{m'}}$ denotes the normalized affinity matrix, $\bar{s}_{ij}$ is the i-th row and j-th column element in $\bar{\mathbf{S}}$. $\bar{\mathbf{S}}$ aims to preserve the semantic neighbors for each node in C_m. If the inner product between node $v_i \in \mathcal{V}^m$ and $v_j \in \mathcal{V}^{m'}$ is greater than $\hat{K}$, we consider v_j to be a neighbor that has similar semantic information to v_i and should be preserved. Next, we download $\hat{\mathbf{H}}_m$ and $\bar{\mathbf{S}}_m$ from the server to client C_m, establishing the new feature matrix $\tilde{\mathbf{H}}^m \in \mathbb{R}^{(N_m+N_{m'})\times d}$ and new adjacency matrix $\tilde{\mathbf{A}}^m \in \mathbb{R}^{(N_m+N_{m'})\times(N_m+N_{m'})}$:

$$\tilde{\mathbf{H}}^m = \begin{bmatrix} \mathbf{H}^m \\ \hat{\mathbf{H}}^{m'} \end{bmatrix}, \quad \tilde{\mathbf{A}}^m = \begin{bmatrix} \mathbf{A}^m & \bar{\mathbf{S}} \\ \bar{\mathbf{S}}^\top & \mathbf{I} \end{bmatrix}, \tag{7}$$

where $\mathbf{I} \in \mathbb{R}^{N_{m'} \times N_{m'}}$ is identity matrix. The reason we deploy $\mathbf{I}$ is that the generation of the affinity matrix $\mathbf{S}$ is based on $\mathbf{H}^{m'}$ without the interaction in $C_{m'}$. Furthermore, topology suture can leverage cross-client affinity nodes to enhance the semantic information. However, the original $\mathbf{A}^{m'}$ of $C_{m'}$ may introduce interference, contaminating the affinity semantics of these suture neighbors.

3.4 Local-Global Consistency Estimation

Since nodes exhibit complex behavior patterns in the intricate topology, we introduce local-global quality scores to comprehensively estimate the consistencies. Local consistencies evaluate the consistency between nodes and their corresponding neighboring patterns. The intuition is that normal nodes should keep consistency with neighbors; however, abnormal nodes usually exhibit high heterophily because of malicious camouflage. Global consistency assesses the node expression in the client, since normal nodes should be closer to the prototype because normal nodes dominate the contributions in the prototypes.

For local consistency estimation, given the client C_m, we regard the latent representation $\mathbf{H}^m$ as local representations since it excludes interactive semantics about neighbors. Then, we obtain the neighboring information $\mathbf{Z} \in \mathbb{R}^{N_m \times d}$ via the message-passing mechanism based on the representation via topology suture and calculate the local consistencies of nodes $\mathbf{s}_L^m \in \mathbb{R}^{N_m}$ based on the inner product:

$$\mathbf{Z}^m = \mathrm{GNN}(\tilde{\mathbf{H}}^m, \tilde{\mathbf{A}}^m), \quad \mathbf{s}_L^m = \mathrm{Sum}(\mathbf{Z}^m \odot \mathbf{H}^m), \tag{8}$$

where $\odot$ is the element-wise product and $\mathrm{Sum}(\cdot)$ is the summation along the dimensions. Then, we calculate the similarity between the prototype and the representation of each node to obtain global consistencies:

$$s_i^m = \mathbf{c}^{m\top} \mathbf{z}_i^m, \tag{9}$$

where $\mathbf{c}^m = \mathrm{Mean}(\mathbf{H}^m)$ denotes prototypes of local representations based on mean pooling, $s_i^m \in \mathbb{R}$ is the global consistency score of node v_i and $\mathbf{z}_i^m = \mathbf{Z}^m[i,:] \in \mathbb{R}^d$ is the representations of node v_i. Next, we obtain the global score vector for all nodes $\mathbf{s}_G^m = [s_1^m, \cdots, s_{N_m}^m] \in \mathbb{R}^{N_m}$. Therefore, the final anomaly scores $\mathbf{s}^m \in \mathbb{R}^{N_m}$ for nodes are formulated as:

$$\mathbf{s}^m = \mathbf{s}_G^m + \mathbf{s}_L^m. \tag{10}$$

To conduct effective unsupervised learning, UnFedGAD uses a threshold K to define the pseudo-labels. Specifically, samples with consistency scores lower than K are considered anomalies. Normal nodes should exhibit high local and global consistencies. We introduce a cross-entropy loss function to learn the scores and a reconstruction loss with an MLP_z to preserve precise node-level features:

$$\mathcal{L}_{un} = -\sum_{i=1}^{N_m} \log s_{(i,+)}^m + \log(1 - s_{(i,-)}^m), \quad \mathcal{L}_{\text{rec}} = ||\mathrm{MLP}_z(\mathbf{Z}^m) - \mathbf{X}||_2, \tag{11}$$

where $s^m_{(i,+)}$ and $s^m_{(i,-)}$ respectively denote pseudo normal samples and pseudo abnormal samples. However, a high pseudo-label noise rate may cause the risk of confirmation bias during early training epochs. Therefore, we introduce a warm-up strategy for the initial E_w epochs, where the model is optimized solely on the structural reconstruction objective (or limited labeled data) to establish a stable feature space.

The total objective function is obtained by integrating the unsupervised contrastive learning with the privacy protection loss function:

$$\mathcal{L} = \mathcal{L}_{\text{un}} + \eta\mathcal{L}_{\text{rec}} + \gamma\mathcal{L}_{\text{pp}}, \tag{12}$$

where γ and η are hyperparameters to control the contribution of different components in the loss function. The overall algorithm is shown in Section N.

3.5 Personalized Bidirectional Updating Mechanism

The personalized bidirectional updating mechanism has two primary contributions compared with traditional federated algorithms. First, traditional federated algorithms lack dynamic weights to evaluate semantic quality in the clients. Instead, we utilize prototype similarity integrated with fixed statistics of clients to dynamically evaluate the contributions of clients. Second, traditional methods directly download the global consensus as local parameters, while the discrepancies between clients, especially in highly non-IID and strong heterogeneity scenarios, prevent the consensus from adapting to client-specific data characteristics. To solve this, we propose the bidirectional updating mechanism, which introduces a weighted client updating process based on prototype semantics instead of directly using global consensus as local parameters.

For the server updating process, although there may be significant discrepancies between clients due to strong heterogeneity, the private graphs in the client stem from the same complete graph; therefore, we assume they follow a latent global prototype. Given prototype set $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_M\}$, we calculate the similarity between $\mathbf{c}_g$ and $\mathbf{c}_m$ to obtain the dynamic weights for client C_m:

$$w_m = \frac{N_m}{N}\mathbf{c}_g{\mathbf{c}_m}^\top, \tag{13}$$

where $\mathbf{c}_g = \frac{1}{M}\sum_{m=1}^{M}\mathbf{c}_m$ is the prototype of all clients, w_m is the affinity between client C_m and the global prototype. Eq. (13) contains two parts: (1) $\frac{N_m}{N}$ indicates the contribution of clients in generating $\mathbf{c}_g$. In common sense, clients with larger sample sizes achieve better generalization performance. (2) $\mathbf{c}_g{\mathbf{c}_m}^\top$ indicates the similarity between the client prototype and the global prototype. A high value usually indicates the client has more robust representations and suggests local models can extract features with more discriminative consensus. Then, we can obtain the affinity vector for all clients $\mathbf{w}_m \in \mathbb{R}^M$. We obtain the normalized $\bar{\mathbf{w}}_m \in \mathbb{R}^M$ using the softmax function to generate the dynamic weights for parameter updating. Denote the local parameters of client C_m and global parameters at t iteration respectively as $\Theta^m_l(t)$ and $\Theta_g(t)$, then the updating process can be formulated:

$$\Theta_g(t+1) = \Theta_g(t) + \sum_{m=1}^{M}\bar{w}_m(\Theta_l(t) - \Theta_g(t)). \tag{14}$$

For the client updating process, directly downloading the global parameters will eliminate the local knowledge learned by the private model, which may impair the detection ability of distributed clients. Therefore, we utilize the parameters from C_m and the nearest client $C_{m'}$, integrated with the global consensus, to generate the updated local parameters. For client C_m, we first calculate the similarity to generate weights for Θ_g and $\Theta^{m'}_l$:

$$w_{mg} = \mathbf{c}_g{\mathbf{c}_m}^\top, w_{mm'} = \mathbf{c}_m{\mathbf{c}_{m'}}^\top. \tag{15}$$

Then, we normalize the weight vector $\mathbf{w}_m = [w_{mg}, w_{mm'}]$ using the softmax function and obtain the normalized weights $\bar{\mathbf{w}}_m = [\bar{w}_{mg}, \bar{w}_{mm'}]$. Finally, the client update process can be formulated:

$$\Theta^m_l(t+1) = \bar{w}_{mg}\Theta_g(t+1) + \bar{w}_{mm'}\Theta^{m'}_l(t) + \alpha\Theta^m_l(t). \tag{16}$$

where α is a hyperparameter.

3.6 THEORETICAL ANALYSIS

First, we formally establish the theoretical privacy guarantees using the information leakage bound. Due to space constraints, the rigorous mathematical proof is provided in Section B.

Theorem 1 (Privacy Leakage Upper Bound). *For any client, given the latent representation H, its encrypted representations $\hat{H}$, and the reconstruction results $\hat{X}$, the privacy leakage upper bound of UnFedGAD is $M_1 - 0.01M2$, where $M_1 = \mathbb{E}[\mathcal{D}_{KL}(p(h|x)|q(h))]$ and $M_2 = H(\hat{X}) - \frac{1}{2}\log(2\pi \exp \mathbb{E}[||\hat{X} - X||_2])$.*

This theorem provides theoretical privacy guarantees, indicating that privacy leakage caused by model training has a strict upper bound, thereby ensuring the basic requirements for individual privacy in a federated environment. In addition, we provide empirical studies in Section C.

Second, we conduct complexity analysis of the topology suture and communication overheads to demonstrate the theoretical scalability and communication efficiency, as shown in Section D.

Table 1: The AUROC and AUPRC performance of UnFedGAD. The best performance is in bold, and the second is underlined. All results are in %.

Metric	Baselines	Dataset							
		Cora	ACM	BlogCatalog	Flickr	ognb-Arxiv	T-Finance	Amazon	Reddit
AUROC	UMGAD	$72.18_{\pm 4.5}$	$65.45_{\pm 3.7}$	$58.36_{\pm 4.2}$	$56.84_{\pm 3.9}$	$54.48_{\pm 3.4}$	$69.37_{\pm 4.1}$	$70.84_{\pm 4.6}$	$53.26_{\pm 3.6}$
	G3AD	$68.36_{\pm 4.1}$	$58.28_{\pm 3.5}$	$58.47_{\pm 4.3}$	$60.46_{\pm 4.0}$	$57.96_{\pm 3.8}$	$68.57_{\pm 3.9}$	$67.16_{\pm 4.4}$	$52.83_{\pm 3.4}$
	CARE	$65.47_{\pm 3.6}$	$60.06_{\pm 3.1}$	$56.83_{\pm 3.8}$	$57.86_{\pm 3.5}$	$55.48_{\pm 3.3}$	$70.94_{\pm 4.0}$	$69.37_{\pm 4.2}$	$51.27_{\pm 3.0}$
	FAGAD	$70.94_{\pm 3.9}$	$59.64_{\pm 3.3}$	$60.18_{\pm 4.0}$	$58.19_{\pm 3.7}$	$60.96_{\pm 3.6}$	$70.41_{\pm 3.8}$	$72.79_{\pm 4.3}$	$52.74_{\pm 3.2}$
	FedPGT	$65.23_{\pm 4.3}$	$62.47_{\pm 3.8}$	$57.16_{\pm 4.5}$	$55.84_{\pm 4.1}$	$59.67_{\pm 3.9}$	$72.29_{\pm 4.4}$	$65.34_{\pm 4.2}$	$52.46_{\pm 3.7}$
	PFGA	$69.64_{\pm 4.4}$	$56.18_{\pm 3.6}$	$59.37_{\pm 4.6}$	$59.46_{\pm 4.2}$	$61.96_{\pm 4.0}$	$68.57_{\pm 3.9}$	$72.16_{\pm 4.7}$	$\underline{54.83}_{\pm 3.5}$
	FedC4	$72.47_{\pm 3.8}$	$61.06_{\pm 3.2}$	$61.83_{\pm 3.9}$	$\underline{62.86}_{\pm 3.7}$	$59.48_{\pm 3.4}$	$72.94_{\pm 4.0}$	$69.37_{\pm 4.1}$	$52.27_{\pm 3.1}$
	FedCAD	$\underline{75.94}_{\pm 3.7}$	$\underline{68.60}_{\pm 3.1}$	$62.15_{\pm 3.8}$	$58.69_{\pm 3.6}$	$\underline{62.96}_{\pm 3.5}$	$\underline{74.52}_{\pm 4.1}$	$73.70_{\pm 4.0}$	$53.74_{\pm 3.2}$
	FedCLGN	$73.81_{\pm 3.5}$	$67.64_{\pm 3.0}$	$\underline{65.18}_{\pm 3.7}$	$61.19_{\pm 3.6}$	$61.23_{\pm 3.4}$	$71.41_{\pm 3.8}$	$\underline{75.79}_{\pm 4.2}$	$52.74_{\pm 3.1}$
	UnFedGAD	$\mathbf{80.23}_{\pm 1.4}$	$\mathbf{76.55}_{\pm 1.2}$	$\mathbf{71.47}_{\pm 1.5}$	$\mathbf{68.02}_{\pm 1.3}$	$\mathbf{70.50}_{\pm 1.2}$	$\mathbf{81.36}_{\pm 1.5}$	$\mathbf{83.10}_{\pm 1.6}$	$\mathbf{56.20}_{\pm 1.1}$
AUPRC	UMGAD	$35.27_{\pm 4.6}$	$18.43_{\pm 3.2}$	$25.10_{\pm 4.1}$	$11.54_{\pm 3.7}$	$10.77_{\pm 2.9}$	$28.05_{\pm 4.3}$	$44.11_{\pm 4.8}$	$9.05_{\pm 3.1}$
	G3AD	$30.54_{\pm 4.2}$	$12.18_{\pm 3.5}$	$22.85_{\pm 4.6}$	$18.30_{\pm 3.9}$	$13.76_{\pm 4.1}$	$26.57_{\pm 3.8}$	$38.54_{\pm 4.7}$	$6.83_{\pm 3.3}$
	CARE	$25.50_{\pm 3.1}$	$15.52_{\pm 2.6}$	$24.86_{\pm 3.4}$	$13.92_{\pm 3.0}$	$11.38_{\pm 2.8}$	$33.58_{\pm 3.7}$	$44.84_{\pm 4.2}$	$5.27_{\pm 2.5}$
	FAGAD	$28.94_{\pm 3.5}$	$13.63_{\pm 2.9}$	$27.09_{\pm 3.8}$	$16.56_{\pm 3.3}$	$19.46_{\pm 3.6}$	$35.10_{\pm 3.9}$	$50.79_{\pm 4.4}$	$6.74_{\pm 2.8}$
	FedPGT	$23.23_{\pm 3.8}$	$\underline{22.38}_{\pm 3.1}$	$20.83_{\pm 4.0}$	$17.97_{\pm 3.6}$	$17.24_{\pm 3.3}$	$39.51_{\pm 4.2}$	$38.05_{\pm 3.9}$	$5.22_{\pm 2.9}$
	PFGA	$29.64_{\pm 4.4}$	$17.34_{\pm 3.6}$	$25.99_{\pm 4.7}$	$16.58_{\pm 3.8}$	$22.54_{\pm 4.1}$	$27.17_{\pm 3.7}$	$47.42_{\pm 4.9}$	$\underline{9.01}_{\pm 3.4}$
	FedC4	$36.47_{\pm 3.9}$	$16.75_{\pm 3.0}$	$27.84_{\pm 3.6}$	$\underline{21.07}_{\pm 3.4}$	$15.68_{\pm 3.1}$	$40.54_{\pm 4.1}$	$42.86_{\pm 3.8}$	$7.27_{\pm 2.7}$
	FedCAD	$\underline{39.56}_{\pm 4.1}$	$21.65_{\pm 3.4}$	$28.45_{\pm 3.9}$	$18.54_{\pm 3.5}$	$\underline{22.56}_{\pm 3.7}$	$\underline{44.76}_{\pm 4.4}$	$48.66_{\pm 4.0}$	$8.82_{\pm 3.0}$
	FedCLGN	$35.94_{\pm 3.7}$	$20.67_{\pm 3.2}$	$\underline{30.85}_{\pm 3.8}$	$20.11_{\pm 3.4}$	$21.36_{\pm 3.6}$	$36.17_{\pm 3.9}$	$\underline{50.34}_{\pm 4.3}$	$7.23_{\pm 2.8}$
	UnFedGAD	$\mathbf{46.26}_{\pm 1.6}$	$\mathbf{30.59}_{\pm 1.4}$	$\mathbf{38.16}_{\pm 1.8}$	$\mathbf{26.62}_{\pm 1.5}$	$\mathbf{30.62}_{\pm 1.4}$	$\mathbf{50.04}_{\pm 1.9}$	$\mathbf{60.84}_{\pm 2.0}$	$\mathbf{10.80}_{\pm 1.3}$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets and Baselines We employ eight public graph anomaly detection datasets to validate the effectiveness of UnFedGAD, including Cora (Tang & Liu, 2009), ACM (Tang & Liu, 2009), BlogCatalog (Tang & Liu, 2009), Flickr (Tang & Liu, 2009), ogbn-Arxiv (Hu et al., 2020a), T-Finance (Zhang et al., 2026), Amazon (McAuley & Leskovec, 2013), and Reddit (Wang et al., 2021). We compared with three families of baselines. The first family is traditional unsupervised GAD methods, including UMGAD (Li et al., 2025a), G3AD (Bei et al., 2025), CARE (Zheng et al., 2025), and FAGAD (Gu et al., 2025). The second family is traditional unsupervised FGL methods, containing FedPGT (Jia et al., 2025), PFGA (Zhang et al., 2025b), and FedC4 (Chen et al., 2025). The third family refers to unsupervised federated GAD methods, involving FedCAD (Kong et al., 2025) and FedCLGN (Wu et al., 2025). More details about datasets and baselines are shown in Section E.

Implementation and Metrics The Adam optimizer with a learning rate of 1e-5 is used for all methods. The hidden dimension d is set to 64. E_t and E_w are set to 500 and 100. The number M is 10. The client-server communication epoch E_c is 50. The hyperparameters α, γ, η, and $\hat{K}$ are, respectively, set to 0.5, 1, 1, and 0.5. K is set to 0.2 for Cora, ACM, BlogCatalog, and Flickr, and 0.3 for ogbn-Arxiv, T-Finance, Amazon, and Reddit. The GNN encoder is set to GIN (Xu et al., 2019) by default. The experiments are conducted 5 times, and we record the mean observations with the standard deviation. We leverage AUROC and AUPRC to evaluate the performance of UnFedGAD. More details are shown in Section F.

4.2 Experimental Analysis

Performance In this section, we validate the effectiveness of UnFedGAD compared with baselines. As shown in Table 1, UnFedGAD achieves the best performance on all datasets, achieving an average improvement of 5.84% and 6.68%, respectively, on the AUROC and AUPRC metrics compared with the runner-up. The reason can be mainly threefold: (1) UnFedGAD utilizes cross-client topology suture to enhance the local semantic propagation and capture more discriminative node features, alleviating the sparsity problem. (2) UnFedGAD employs a pseudo-label-based cross-entropy loss function to distinguish normal and abnormal nodes, reducing reliance on labels while improving traditional unsupervised learning ability. (3) UnFedGAD develops a novel personalized updating mechanism for federated clients based on the similarity of prototypes, adaptively obtaining high-quality global consensus while preserving local properties, moderating heterogeneity impacts.

Ablation Study We introduce three variants to validate the effectiveness of the primary components in UnFedGAD. From the observations in Table 2, we can draw several conclusions. First, cross-client topology suture helps to improve the performance of each client because of the key semantic information between clients. Second, reconstruction loss can improve the regularization of the GNN encoders for node features. Third, the IB loss in the privacy protection mechanism protects the original features in the local client, while enhancing the semantics of the uploaded representations. More ablations on consistency estimation and the warm-up strategy are shown in Section G.

Table 2: The results of the ablation study. Best results are in bold. All results are in %.

Metric	Variants	Dataset							
		Cora	ACM	BlogCatalog	Flickr	ogbn-Arxiv	T-Finance	Amazon	Reddit
AUROC	UnFedGAD$_{\text{PP}}$	60.02$_{\pm 2.3}$	52.86$_{\pm 2.5}$	52.84$_{\pm 2.1}$	46.38$_{\pm 2.4}$	54.92$_{\pm 2.6}$	66.68$_{\pm 1.8}$	63.24$_{\pm 2.5}$	50.18$_{\pm 2.3}$
	UnFedGAD$_{\text{REC}}$	73.46$_{\pm 2.2}$	66.14$_{\pm 3.1}$	60.02$_{\pm 2.3}$	49.86$_{\pm 2.2}$	64.78$_{\pm 1.3}$	70.54$_{\pm 2.5}$	67.92$_{\pm 2.2}$	54.76$_{\pm 2.4}$
	UnFedGAD$_{\text{IB}}$	76.18$_{\pm 2.0}$	70.72$_{\pm 1.9}$	66.96$_{\pm 2.1}$	56.12$_{\pm 3.0}$	67.42$_{\pm 2.1}$	73.96$_{\pm 2.3}$	72.48$_{\pm 1.8}$	55.38$_{\pm 2.3}$
	UnFedGAD	**80.23**$_{\pm 1.4}$	**76.55**$_{\pm 1.2}$	**71.47**$_{\pm 1.5}$	**68.02**$_{\pm 1.3}$	**70.50**$_{\pm 1.2}$	**81.36**$_{\pm 1.5}$	**83.10**$_{\pm 1.6}$	**56.20**$_{\pm 1.1}$
AUPRC	UnFedGAD$_{\text{PP}}$	28.14$_{\pm 2.5}$	12.04$_{\pm 2.3}$	18.72$_{\pm 2.7}$	8.46$_{\pm 2.4}$	13.86$_{\pm 2.5}$	28.92$_{\pm 2.2}$	41.18$_{\pm 2.6}$	7.84$_{\pm 2.6}$
	UnFedGAD$_{\text{REC}}$	41.28$_{\pm 2.3}$	19.18$_{\pm 2.6}$	24.06$_{\pm 2.0}$	12.62$_{\pm 2.1}$	20.48$_{\pm 2.8}$	39.14$_{\pm 2.6}$	48.96$_{\pm 2.4}$	9.62$_{\pm 2.1}$
	UnFedGAD$_{\text{IB}}$	43.92$_{\pm 2.6}$	25.84$_{\pm 1.9}$	30.48$_{\pm 2.3}$	18.96$_{\pm 2.0}$	25.86$_{\pm 2.3}$	44.88$_{\pm 2.4}$	52.12$_{\pm 2.3}$	10.24$_{\pm 2.3}$
	UnFedGAD	**46.26**$_{\pm 1.6}$	**30.59**$_{\pm 1.4}$	**38.16**$_{\pm 1.8}$	**26.62**$_{\pm 1.5}$	**30.62**$_{\pm 1.4}$	**50.04**$_{\pm 1.9}$	**60.84**$_{\pm 2.0}$	**10.80**$_{\pm 1.3}$

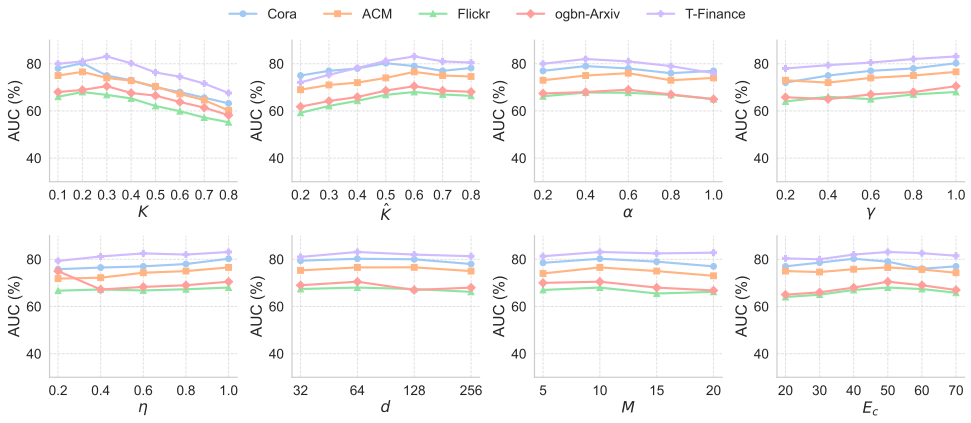

Figure 2: The sensitivity performance of eight hyperparameters on five datasets regarding AUROC.

Sensitivity In this section, we explore the influence of hyperparameters, including K, $\hat{K}$, α, γ, η, d, M, and E_c, as shown in Fig. 2. Specifically, K affects performance by controlling the training process of local and global anomaly estimation. $\hat{K}$ affects the quality of representation learning by selecting cross-client semantic neighbors of different sizes. α influences global consensus by adjusting the weights of local knowledge when updating client parameters. Changing the weights of the learning objective in the loss function, that is, γ and η, leads to performance fluctuations. d has relatively stable performance compared to others. The number of clients M causes

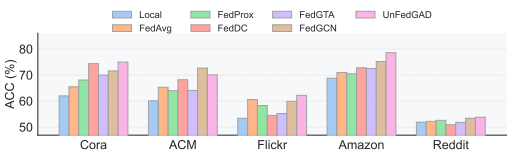

Figure 3: The AUROC results of the personalized updating mechanism experiments on five datasets.

discrepancies in the sample size of each client, which leads to a performance change. Communication epochs can determine the quality of the global consensus, which also causes a deviation in learning ability.

Utility of Personalized Updating Mechanism To validate the effectiveness of the personalized updating mechanism, we first introduce six variants, including Local, FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedDC (Gao et al., 2022), FedGTA (Li et al., 2023), and FedGCN (Yao et al., 2023), where Local means without server-client parameter updating. As shown in Fig. 3, we can see that the proposed personalized updating mechanism achieves superior performance. The reason is twofold: First, it leverages prototypes to evaluate the qualities of clients and generate dynamic weights for parameter integration. Second, it partially preserves the local knowledge when downloading the global consensus to the client. More experiments are shown in Section I.

Table 3: The AUROC evaluation of the components in the personalized updating mechanism (Eq. (14) and Eq. (16)) on four datasets. Best results are in bold. All results are in %.

Variants	Dataset			
	Cora	ACM	BlogCatalog	Flickr
UnFedGAD$_{\text{g(t)}}$	75.92$_{\pm 2.4}$	68.13$_{\pm 2.7}$	66.02$_{\pm 2.9}$	63.84$_{\pm 2.1}$
UnFedGAD$_{\text{l(t)-g(t)}}$	74.36$_{\pm 2.9}$	67.48$_{\pm 2.5}$	65.91$_{\pm 3.1}$	62.72$_{\pm 2.6}$
UnFedGAD$_{\text{avg}}$	76.08$_{\pm 2.6}$	69.02$_{\pm 3.0}$	67.34$_{\pm 2.4}$	64.96$_{\pm 2.2}$
UnFedGAD$_{m'}$	77.42$_{\pm 2.1}$	71.08$_{\pm 2.3}$	68.02$_{\pm 2.5}$	64.28$_{\pm 1.9}$
UnFedGAD$_{mm'}$	77.16$_{\pm 2.7}$	70.64$_{\pm 2.1}$	68.46$_{\pm 2.2}$	64.02$_{\pm 2.4}$
UnFedGAD$_{w}$	76.88$_{\pm 2.3}$	70.92$_{\pm 2.6}$	68.14$_{\pm 2.8}$	64.36$_{\pm 2.0}$
UnFedGAD	**80.23**$_{\pm 1.4}$	**76.55**$_{\pm 1.2}$	**71.47**$_{\pm 1.5}$	**68.02**$_{\pm 1.3}$

Second, we evaluate the components of the personalized updating mechanism by introducing six variants. From Table 3, for the server updating process (Eq. (14)), $\Theta_g(t)$, $\Theta_l(t)-\Theta_g(t)$, and adaptive weights $\bar{w}_m$ ameliorate the global consensus, exhibiting varying improvements in performance. For client updating process, $\bar{\mathbf{w}}_m$, $\Theta_l^{m'}(t)$ and $\Theta_l^m(t)$ significantly facilitate the discrimination of local knowledge. The variant details and more experimental results are shown in Section I.

Non-IID Challenges In this study, we generate distributed clients using the same graph datasets, which undoubtedly assumes that each local dataset conforms to a uniform potential distribution. To validate the performance of the model in strong non-IID heterogeneity, we set up five clients, each containing training nodes from Cora, ACM, Flickr, T-Finance, and Amazon, respectively, with approximately the same number of nodes per client. This setup establishes a highly non-IID environment. For performance evaluation, we first test the models of the five clients using Cora test nodes, and then sequentially test them using the other datasets. As shown in Table 4, although the observations exhibit degradation due to the strong heterogeneity between clients, UnFedGAD maintains superior performance over baselines, demonstrating effective resistance to strong heterogeneity.

Table 4: The AUROC(%) performance of Non-IID Challenges with five datasets. Best results are in bold. All results are in %.

Models	Datasets				
	Cora	ACM	Flickr	T-Finance	Amazon
PFGA	68.54$_{\pm 2.5}$	60.73$_{\pm 2.6}$	55.45$_{\pm 3.0}$	65.48$_{\pm 2.0}$	65.37$_{\pm 2.2}$
FedC4	64.89$_{\pm 2.8}$	65.15$_{\pm 2.6}$	53.27$_{\pm 3.0}$	61.69$_{\pm 2.5}$	68.10$_{\pm 2.0}$
FedCAD	70.08$_{\pm 2.7}$	66.54$_{\pm 2.9}$	55.83$_{\pm 2.5}$	69.41$_{\pm 2.3}$	66.07$_{\pm 2.2}$
FedCLGN	67.27$_{\pm 2.0}$	62.42$_{\pm 2.4}$	57.84$_{\pm 2.4}$	68.33$_{\pm 2.0}$	70.09$_{\pm 2.0}$
UnFedGAD	**73.05**$_{\pm 1.5}$	**70.24**$_{\pm 1.1}$	**61.38**$_{\pm 1.6}$	**75.41**$_{\pm 1.2}$	**72.88**$_{\pm 2.0}$

Additional Experiments Section G shows the ablation studies of local and global consistency learning. Section H studies the sensitivity of the threshold across unseen datasets. Section J explores the performance fluctuation of using different graph encoders. Section K demonstrates that UnFedGAD has powerful robustness against nodes with noisy information. Section L provides more intuitive statistical information to demonstrate the impact of topology suture on nodes in each client. Section M provides empirical validation to demonstrate the resistance against noise propagation.

5 Conclusion

This paper explores sparsity and heterogeneity issues in federated GAD and proposes UnFedGAD, which aims to explore an unsupervised solution that supplements local semantics through a privacy-preserving cross-client topology suture and alleviates heterogeneity issues via evaluating prototype-based weights for clients and introducing bidirectional parameter updating. Additionally, UnFedGAD enhances local training by pseudo-labels generated based on local-global consistency estimation. Extensive experiments on eight public datasets demonstrate the effectiveness and advancement of UnFedGAD compared with nine baselines. In the future, we plan to explore more advanced learning paradigms, such as adversarial and prompt learning, to capture more discriminative features for federated GAD.

AI USE STATEMENT

During manuscript preparation, the authors used ChatGPT solely for grammatical correction and English-language polishing. The tool was not used for research design, methodological development, code generation, experimentation, data analysis, or the generation of scientific results or conclusions. The authors take full responsibility for the content, accuracy, and integrity of the manuscript.

ETHICS STATEMENT

This study does not involve human subjects, practices related to dataset releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues (e.g., IRB, documentation, research ethics).

REPRODUCIBILITY STATEMENT

Section 3 defines the problem, the proposed framework, and the corresponding objective. The supplement specifies more details about experimental configuration and results. All evaluated datasets are public and cited; no private data are used.

REFERENCES

Yuanchen Bei, Sheng Zhou, Jinke Shi, Yao Ma, Haishuai Wang, and Jiajun Bu. Guarding graph neural networks for unsupervised graph anomaly detection. *IEEE Trans. Neural Networks Learn. Syst.*, 36(9):16840–16853, 2025.

Jinyu Cai, Yunhe Zhang, Jicong Fan, and See-Kiong Ng. LG-FGAD: an effective federated graph anomaly detection framework. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3760–3769, 2024a.

Jinyu Cai, Yunhe Zhang, Zhoumin Lu, Wenzhong Guo, and See-Kiong Ng. Towards effective federated graph anomaly detection via self-boosted knowledge distillation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5537–5546, 2024b.

Zekai Chen, Xunkai Li, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Rethinking client-oriented federated graph learning. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 393–402, 2025.

Xingbo Fu, Zihan Chen, Yinhan He, Song Wang, Binchi Zhang, Chen Chen, and Jundong Li. Virtual nodes can help: Tackling distribution shifts in federated graph learning. In *the Association for the Advancement of Artificial Intelligence*, pp. 16657–16665, 2025.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pp. 10102–10111, 2022.

Sayan Ghosh, Mahantesh Halappanavar, Antonino Tumeo, Ananth Kalyanaraman, Hao Lu, Daniel G. Chavarría-Miranda, Arif Khan, and Assefaw Hadish Gebremedhin. Distributed louvain algorithm for graph community detection. In *2018 IEEE International Parallel and Distributed Processing Symposium*, pp. 885–895, 2018. doi: 10.1109/IPDPS.2018.00098.

Ming Gu, Gaoming Yang, Zhuonan Zheng, Meihan Liu, Haishuai Wang, Jiawei Chen, Sheng Zhou, and Jiajun Bu. Frequency self-adaptation graph neural network for unsupervised graph anomaly detection. *Neural Networks*, 190:107612, 2025.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 1024–1034, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020a.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations*, 2020b.

Tairan Huang, Qiutong Li, Cong Xu, Jianliang Gao, Zhao Li, and Shichao Zhang. Revisiting low-homophily for graph-based fraud detection. *Neural Networks*, 188:107407, 2025.

Haohe Jia, Yi Huang, Hongbin Zhu, and Hongfeng Chai. Personalized graph transformer for federated graph learning. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2025.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017.

Xiangjie Kong, Wenyi Zhang, Hui Wang, Mingliang Hou, Xin Chen, Xiaoran Yan, and Sajal K. Das. Federated graph anomaly detection via contrastive self-supervised learning. *IEEE Trans. Neural Networks Learn. Syst.*, 36(5):7931–7944, 2025.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the Third Conference on Machine Learning and Systems*, 2020.

Xiang Li, Jianpeng Qi, Zhongying Zhao, Guanjie Zheng, Lei Cao, Junyu Dong, and Yanwei Yu. UMGAD: unsupervised multiplex graph anomaly detection. In *41st IEEE International Conference on Data Engineering*, pp. 3724–3737, 2025a.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Ronghua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. *Proc. VLDB Endow.*, 17(1):41–50, 2023.

Xunkai Li, Yinlin Zhu, Boyang Pang, Guochen Yan, Yeyu Yan, Zening Li, Zhengyu Wu, Wentao Zhang, Ronghua Li, and Guoren Wang. Openfgl: A comprehensive benchmark for federated graph learning. *Proc. VLDB Endow.*, 18(5):1305–1320, 2025b.

Yuqi Li, Guosheng Zang, Chunyao Song, and Xiaojie Yuan. A universal adaptive algorithm for graph anomaly detection. *Inf. Process. Manag.*, 62(1):103905, 2025c.

Zhengyang Liu, Hang Yu, and Xiangfeng Luo. Federated graph anomaly detection via disentangled representation learning. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney*, pp. 1216–1224, 2025.

Julian J. McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *22nd International World Wide Web Conference*, pp. 897–908, 2013.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, 2017.

Xinjun Pei, Xiaoheng Deng, Shengwei Tian, Ping Jiang, Yunlong Zhao, and Kaiping Xue. A privacy-preserving graph neural network for network intrusion detection. *IEEE Trans. Dependable Secur. Comput.*, 22(1):740–756, 2025.

Zihan Tan, Suyuan Huang, Guancheng Wan, Wenke Huang, He Li, and Mang Ye. S2FGL: spatial spectral federated graph learning. In *Forty-second International Conference on Machine Learning*, 2025.

Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 817–826, 2009.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations*, 2018.

Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1239–1248, 2021.

Nannan Wu, Yazheng Zhao, Hongdou Dong, Keao Xi, Wei Yu, and Wenjun Wang. Federated graph anomaly detection through contrastive learning with global negative pairs. In *the Association for the Advancement of ArtificialIntelligence*, pp. 21554–21562, 2025.

Chunjing Xiao, Yao Tong, Jiahui Lu, Yuxia Xue, Wenxin Tai, Zhangtao Cheng, and Fan Zhou. Graph-enhanced multi-scale contrastive learning for graph anomaly detection with adaptive diffusion models. *IEEE Trans. Big Data*, 11(6):2946–2957, 2025.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations*, 2019.

Yuhang Yao, Weizhao Jin, Srivatsan Ravi, and Carlee Joe-Wong. Fedgcn: Convergence-communication tradeoffs in federated training of graph convolutional networks. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023.

Qi Yuan, Yang Liu, Yateng Tang, Xinhuan Chen, Xuehao Zheng, Qing He, and Xiang Ao. Dynamic graph learning with static relations for credit risk assessment. In *the Association for the Advancement of Artificial Intelligence*, pp. 13133–13141, 2025.

Linan Yue, Qi Liu, Yawen Li, Fangzhou Yao, Weibo Gao, and Junping Du. Empowering federated graph rationale learning with latent environments. In *Proceedings of the ACM on Web Conference 2025*, pp. 3863–3873, 2025.

Hao Zhang, Xunkai Li, Yinlin Zhu, and Lianglin Hu. Rethinking federated graph learning: A data condensation perspective. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6967–6975, 2025a.

Liang Zhang, Tao Long, Yang Liu, Lei Zhang, Laizhong Cui, and Qingjiang Shi. Generalizing personalized federated graph augmentation via min-max adversarial learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1924–1935, 2025b.

Tianlong Zhang, Xiaoxi He, Yuxiang Wang, Yi Xu, Rendi Wu, Zhifei Wang, and Yongxin Tong. Fedmetro: Efficient metro passenger flow prediction via federated graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5215–5224, 2025c.

Wenxin Zhang and Cuicui Luo. GE-GNN: gated edge-augmented graph neural network for fraud detection. *IEEE Trans. Big Data*, 11(4):1664–1676, 2025.

Wenxin Zhang and Cuicui Luo. Mclast: Multi-hierarchy contrastive learning graph anomaly detection with structure-awareness. *Neurocomputing*, 669:132480, 2026.

Wenxin Zhang, Jingxing Zhong, Guangzhen Yao, Renda Han, Xiaojian Lin, Lei Jiang, Zeyu Zhang, and Cuicui Luo. Dual-channel heterophilic message passing for graph fraud detection. In *International Joint Conference on Neural Networks*, pp. 1–8, 2025d.

Wenxin Zhang, Yuhao Zhu, Cuicui Luo, Desheng Wu, Xi Xuan, Ljupco Kocarev, Renda Han, and Xiangxiang Lang. Feature and structural uncertainty modeling for risk identification in supply chain networks. *Chaos, Solitons & Fractals*, 201:117238, 2025e.

Wenxin Zhang, Rong Fu, Xi Xuan, Renda Han, Zonghao Ying, Guangzhen Yao, Simon Fong, Lingfei Ren, and Cuicui Luo. A gnn-based fraud detection approach against heterophily inconsistencies. *Appl. Soft Comput.*, 203:116010, 2026. doi: 10.1016/J.ASOC.2026.116010.

Lecheng Zheng, John R. Birge, Haiyue Wu, Yifang Zhang, and Jingrui He. Cluster aware graph anomaly detection. In *Proceedings of the ACM on Web Conference 2025*, pp. 1771–1782, 2025.

A Related Work

In this section, we detail the difference between the proposed node-level federated GAD and the existing graph-level federated GAD. The fundamental difference between them is distinct: Graph-level methods treat graphs as samples instead of nodes. Therefore, the graph structure is completely preserved. However, we focus on node-level federated GAD, which needs to split a complete graph into several subgraphs, inevitably leading to critical interactive topology being missing.

B Theoretical Analysis

In this section, we provide rigorous mathematical proofs for the theoretical privacy guarantees for UnFedGAD.

Theorem 1 (Privacy Leakage Upper Bound). *For any client, given the latent representation H, its encrypted representations $\hat{H}$, and the reconstruction results $\hat{X}$, the privacy leakage upper bound of UnFedGAD is $M_1 - 0.01M2$, where $M_1 = \mathbb{E}[\mathcal{D}_{KL}(p(\hat{h}|x)|q(\hat{h}))]$ and $M_2 = H(\hat{X}) - \frac{1}{2}\log(2\pi \exp \mathbb{E}[||\hat{X} - X||_2])$.*

Proof. First, we infer the upper bound of $I(X, \hat{H})$. From the definition of the information bottleneck, we have:

$$I(X, \hat{H}) = \mathbb{E}_x \int p(\hat{h}|x) \ln \frac{p(\hat{h}|x)}{p(\hat{h})} d\hat{h}.$$

Due to p is intractable, we introduce q, then we have the following equation:

$$I(X, \hat{H}) = \mathbb{E}_x \int p(\hat{h}|x) \ln \frac{p(\hat{h}|x)q(\hat{h})}{p(\hat{h})} q(\hat{h}) d\hat{h}.$$

Therefore, we have the following inequality based on variational inference:

$$I(X, \hat{H}) \leq \mathbb{E}_x[\mathcal{D}_{KL}(p(\hat{h}|x)|q(\hat{h}))].$$

According to the Data Processing Inequality, for any f-divergence, adding noise to the latent variable will only make the distribution closer; then we have:

$$\mathbb{E}_x[\mathcal{D}_{KL}(p(\hat{h}|x)|q(\hat{h}))] \leq \mathbb{E}_x[\mathcal{D}_{KL}(p(h|x)|q(h))].$$

Therefore, we have:

$$I(X, \hat{H}) \leq \mathbb{E}_x[\mathcal{D}_{KL}(p(h|x)|q(h))]$$

Second, we infer the lower bound of $I(\hat{X}, \hat{H})$. According to the definition of mutual information, we have:

$$I(\hat{X}, \hat{H}) = H(\hat{X}) - H(\hat{X}|\hat{H}).$$

Given the principle of maximum entropy, that is:

$$H(\hat{X}|\hat{H}) \leq \frac{1}{2}\log(2\pi \exp \mathbb{E}[||\hat{X} - X||_2]),$$

we have the following inequality:

$$I(\hat{X}, \hat{H}) \geq H(\hat{X}) - \frac{1}{2}\log(2\pi \exp \mathbb{E}[||\hat{X} - X||_2]).$$

Let $M_1 = \mathbb{E}[\mathcal{D}_{KL}(p(h|x)|q(h))]$ and $M_2 = H(\hat{X}) - \frac{1}{2}\log(2\pi \exp \mathbb{E}[||\hat{X} - X||_2])$, the final lower bound is formulated as:

$$\mathcal{L}_{\text{IB}} \leq M_1 - 0.01M2.$$

C Privacy Leakage Evaluation and Differential Privacy Guarantees

Although UnFedGAD incorporates Gaussian reparameterization and an information bottleneck (IB) objective to suppress sensitive information in shared representations, these mechanisms alone do not establish formal differential privacy (DP). In particular, the encrypted node representations $\hat{\mathbf{H}}^m$ and the cross-client neighboring features may still expose sensitive information to an adversary. To investigate these risks, we introduce two complementary evaluations: (i) empirical privacy attacks against shared representations and (ii) a differentially private extension of UnFedGAD with an explicit privacy budget.

Threat Model. We consider an honest-but-curious adversary that follows the prescribed communication protocol but attempts to infer private information from the received representations. The adversary may be either the central server or a neighboring client participating in topology suture. Following the communication protocol of UnFedGAD, the adversary observes the encrypted node representations $\hat{\mathbf{H}}^m$, client prototypes $\hat{\mathbf{c}}^m$, and the corresponding cross-client affinity information. For a conservative evaluation, we additionally assume that the adversary possesses auxiliary graph data drawn from a similar distribution and has access to the architecture of the feature encoder. The adversary attempts to reconstruct the original node attributes, infer sensitive node properties, or recover the private graph structure. However, the raw feature matrix $\mathbf{X}^m$ and local adjacency matrix $\mathbf{A}^m$ remain inaccessible. We evaluate the resulting privacy risks under both the original IB-based mechanism and its differentially private extension.

Experiment 1: Representation Reconstruction and Attribute Inference. We first investigate whether the uploaded encrypted representations retain sufficient information to reconstruct private node attributes. Specifically, an adversarial decoder f_ϕ is trained on an auxiliary dataset to approximate the inverse mapping from the shared latent space to the original feature space. Its optimization objective is formulated as

$$\mathcal{L}_{\text{inv}} = \frac{1}{|\mathcal{V}_{\text{aux}}|} \sum_{i \in \mathcal{V}_{\text{aux}}} \left\| f_\phi(\hat{\mathbf{h}}_i) - \mathbf{x}_i \right\|_2^2, \tag{17}$$

where $\mathcal{V}_{\text{aux}}$ denotes the auxiliary node set. After training, the decoder is applied to the encrypted representations of the target client without accessing its original features.

To further evaluate attribute leakage, we train an MLP-based adversarial classifier g_ψ to predict node attributes from the uploaded representations:

$$\mathcal{L}_{\text{attr}} = -\frac{1}{|\mathcal{V}_{\text{aux}}|} \sum_{i \in \mathcal{V}_{\text{aux}}} \sum_{k=1}^{K_a} y_{ik}^a \log g_\psi(\hat{\mathbf{h}}_i)_k, \tag{18}$$

where K_a denotes the number of attribute categories and y_{ik}^a indicates the corresponding attribute label. For datasets without explicitly designated sensitive attributes, we use available categorical node attributes as inference targets.

In addition, we consider a structural reconstruction attack. The adversary employs a graph inversion decoder to estimate the private adjacency matrix:

$$\hat{\mathbf{A}}^m = \sigma\left(f_{\text{str}}(\hat{\mathbf{H}}^m) f_{\text{str}}(\hat{\mathbf{H}}^m)^\top \right), \tag{19}$$

where $\sigma(\cdot)$ is the sigmoid function. The structural decoder is trained on auxiliary graphs with known adjacency matrices and evaluated on the target client's private graph.

We report feature reconstruction MSE, attribute inference AUROC, and edge reconstruction AUROC. Higher reconstruction errors and lower attack AUROC indicate reduced information leakage. For comparison, we evaluate raw latent representations, Gaussian reparameterization without IB, the original UnFedGAD, and its differentially private extension.

Experiment 2: Differentially Private Representation Sharing. To provide a formal privacy guarantee, we extend the original representation-sharing mechanism with node-level clipping and

Table 5: Privacy leakage evaluation under different representation sharing mechanisms. Reconstruction MSE is better when higher, whereas attack AUROC is better when lower. Entries marked "–" are to be filled with measured results.

Sharing Mechanism	Feature MSE ↑	Attribute AUC ↓	Edge AUC ↓
Raw latent representations	0.36	0.78	0.73
Gaussian reparameterization	0.51	0.73	0.71
UnFedGAD w/o IB	0.55	0.72	0.69
UnFedGAD	0.79	0.58	0.64
UnFedGAD-DP	0.84	0.53	0.56

Gaussian perturbation. Specifically, given the latent representation $\mathbf{h}_i^m$ generated by the local feature encoder, we first bound its ℓ_2 norm:

$$\bar{\mathbf{h}}_i^m = \frac{\mathbf{h}_i^m}{\max\left(1, \frac{\|\mathbf{h}_i^m\|_2}{C}\right)}, \tag{20}$$

where C is the clipping threshold.

We then perturb each clipped representation using an independent Gaussian mechanism:

$$\tilde{\mathbf{h}}_i^m = \bar{\mathbf{h}}_i^m + \boldsymbol{\xi}_i, \qquad \boldsymbol{\xi}_i \sim \mathcal{N}\left(\mathbf{0}, \sigma_{\mathrm{DP}}^2 C^2 \mathbf{I}\right). \tag{21}$$

Unlike the learnable noise scale in the original reparameterization mechanism, σ_{DP} is explicitly calibrated according to the desired privacy budget. The perturbed representations $\tilde{\mathbf{H}}^m$ replace $\hat{\mathbf{H}}^m$ in all cross-client communication operations, including prototype generation, neighboring-client selection, affinity calculation, and topology suture.

For the privacy analysis, we adopt a replace-one-node adjacency relation: two local feature datasets are adjacent if they differ in the attributes of exactly one node. We condition on the local adjacency matrix and a fixed feature encoder, and consider one representation-release event. Under this definition, the ℓ_2 sensitivity of the clipped representation release is bounded by

$$\Delta_2 = \max_{D \sim D'} \|f(D) - f(D')\|_2 \le 2C, \tag{22}$$

where $D \sim D'$ denotes adjacent datasets.

According to the Gaussian mechanism, the representation release satisfies (ε, δ)-differential privacy when

$$\sigma_{\mathrm{DP}} \ge \frac{2\sqrt{2\ln(1.25/\delta)}}{\varepsilon}, \qquad 0 < \varepsilon < 1. \tag{23}$$

Consequently, for any pair of adjacent datasets D, D' and any measurable set of possible outputs $\mathcal{S}$,

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \le e^{\varepsilon} \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta, \tag{24}$$

where $\mathcal{M}$ denotes the differentially private representation-release mechanism.

Importantly, client prototypes and cross-client affinity matrices computed exclusively from these perturbed representations are post-processing operations and therefore incur no additional privacy loss for the same release. Similarly, downloading perturbed neighboring representations does not increase the privacy cost of that release.

For multiple releases, we account for privacy loss across communication rounds. If each of the T rounds satisfies $(\varepsilon_t, \delta_t)$-DP, sequential composition yields

$$\varepsilon_{\mathrm{total}} = \sum_{t=1}^{T} \varepsilon_t, \qquad \delta_{\mathrm{total}} = \sum_{t=1}^{T} \delta_t. \tag{25}$$

A tighter total privacy budget can be obtained using Rényi differential privacy accounting.

The above guarantee applies to the feature-release mechanism under the stated fixed-encoder and fixed-topology assumptions. To obtain an end-to-end node-level DP guarantee for the complete federated training protocol, local model updates must also be privatized, and their privacy costs must be included in the composition accountant. In addition, protecting the existence or incident edges of a node requires a separate graph-adjacency privacy analysis.

Table 6: Privacy–utility trade-off of UnFedGAD-DP under different privacy budgets. AUROC and AUPRC are reported as percentages. Entries marked "–" require experimental measurements.

Method	ε	**AUROC** ↑	**AUPRC** ↑	**Feature MSE** ↑
UnFedGAD	N/A	77.85	42.87	0.76
UnFedGAD-DP	0.5	70.12	36.20	2.45
UnFedGAD-DP	1.0	72.45	38.50	1.80
UnFedGAD-DP	2.0	74.80	40.10	1.35
UnFedGAD-DP	4.0	76.50	41.90	0.98
UnFedGAD-DP	8.0	77.30	42.50	0.82

Privacy-Utility Trade-off. We evaluate the differentially private extension using $\varepsilon \in \{0.5, 1, 2, 4, 8\}$ with a fixed $\delta = 10^{-5}$, subject to the dataset size and the corresponding privacy accounting assumptions. For each privacy budget, the noise multiplier is calibrated using the Gaussian mechanism and the composition accountant. We retain the original UnFedGAD architecture and training configuration, changing only the privacy-related operations required by the DP extension. We conduct experiments on Cora, ACM, Flickr, T-Finance, and Amazon, covering different graph sizes, feature distributions, and anomaly patterns. Following the original experimental protocol, each configuration is repeated five times, and the mean and standard deviation are reported. We evaluate anomaly detection performance using AUROC and AUPRC. Meanwhile, we report reconstruction MSE to characterize empirical information leakage. The resulting privacy–utility trade-off is summarized in Table 6.

The evaluation examines whether the proposed topology suture mechanism can retain useful cross-client semantic information under different privacy constraints. In particular, stronger privacy protection introduces larger perturbations, potentially affecting prototype similarity, affinity-based neighbor selection, and subsequent anomaly detection. The experiments quantify this trade-off rather than assuming that information bottleneck regularization or stochastic reparameterization alone eliminates privacy leakage.

D Complexity and Communication Analysis

Complexity Theoretically, for each client C_m, computing its nearest neighboring client $C_{m'}$ requires performing one similarity calculation with each of the other clients, resulting in a computational complexity of $\mathcal{O}(M-1)$, where M denotes the number of clients. Then, matrix multiplication is used to generate the affinity matrix, with a computational complexity of $\mathcal{O}(N_m N_{m'} d)$, where N_m and $N_{m'}$ respectively denote the number of nodes in C_m and $C_{m'}$ and d is the feature dimension. It can be seen that this strategy ensures a linear relationship with respect to any parameter, achieving low computational complexity.

Communication In theory, the communication process of UnFedGAD is more complex than that of traditional FedAvg. In addition to model parameter exchange, it involves uploading prototypes and node representations, as well as downloading cross-client data.

(1) Components. In each communication round t, the total communication overhead S_{total} for a single client C_m with the server consists of the following three parts:

- Parameter Exchange: The client downloads the global parameters Θ_g and uploads the local parameters Θ_l.
- Prototype Exchange: The client uploads its local prototypes $\hat{\mathbf{c}}^m$ and the server broadcasts the global prototype $\mathbf{c}_g$
- Topology Suture Data Exchange: The client needs to upload the encrypted representations $\hat{\mathbf{H}}^m$ and download the representations $\mathbf{H}^{m'}$ of the nearest neighbor client $C_{m'}$.

(2) Notations. In the following, we list the symbol description in Table 7.

Table 7: The symbol description.

Symbol	Description
$\|\Theta\|$	Model parameter size
d	Dimension of the embedding vector
N_m	Number of nodes in the client
$N_{m'}$	Number of nodes in the neighbor client
b	Number of bytes occupied by numerical precision

(3) Formulation. For client C_m, the communication overhead per round is:

$$C_{round} = 2 \cdot |\Theta| \cdot b + 2 \cdot d \cdot b + (N_m + N_{m'}) \cdot d \cdot b,$$

where $N_m \cdot d \cdot b$ corresponds to uploading the local encrypted representation $\hat{\mathbf{H}}^m$, and $N_{m'} \cdot d \cdot b$ corresponds to downloading the representation $\mathbf{H}^{m'}$ of the neighbor client from the server.

The total communication overhead across M clients over T_s rounds is:

$$C_{total} = T_s \times \sum_{m=1}^{M} C_{round}.$$

E Datasets and Baselines

E.1 Datasets

The details of the datasets are as follows:

- **Cora** and **ACM** are academic citation networks, where nodes correspond to published papers and edges denote citation relationships among them. Each paper is associated with an attribute vector represented as a bag-of-words.
- **BlogCatalog** is a social blogging platform, in which nodes represent users and edges capture the follower–follower relationships between them. Node attributes are composed of user-defined tag lists.
- **Flickr** is an online image hosting and sharing platform. Each node corresponds to a user, and edges represent the following relationships among users. The node features consist of both user tags and photo-related tags.
- **ogbn-Arxiv** is a large-scale citation network of academic publications, in which nodes correspond to papers and edges denote citation relationships. Each paper is associated with a 128-dimensional feature vector, constructed by averaging the word embeddings of its title and abstract. The word embeddings are obtained by applying the skip-gram model to the Microsoft Academic Graph (MAG) corpus.
- **T-Finance** is designed for identifying anomalous accounts in transaction networks. Each node represents a unique anonymized account and is associated with a 10-dimensional feature vector capturing information related to registration duration, login behavior, and interaction frequency. Edges in the graph indicate the existence of transaction records between pairs of accounts. Nodes are annotated by human experts as anomalies if they belong to illicit categories such as fraud, money laundering, or online gambling.
- **Amazon** The Amazon dataset consists of user-generated product reviews. Users who receive more than 80% helpful votes are labeled as benign, while those with fewer than 20% helpful votes are regarded as fraudulent entities. To mitigate label leakage, this feature is removed when constructing node representations, resulting in 24-dimensional node features. The constructed graph incorporates three types of relations: (1) U–P–U, which connects users who have reviewed at least one common product; (2) U–S–U, which links users who assign the same star rating within a one-week time window; and (3) U–V–U, which connects users whose review texts exhibit mutual similarity within the top 5%.

- **Reddit** is a heterogeneous user–subreddit interaction graph collected from the Reddit platform. The dataset includes the 1,000 most active subreddits and the 10,000 most active users as nodes. Each user node is assigned a binary label indicating whether the user has been banned, with banned users regarded as anomalies. Textual content from posts is transformed into feature representations, and the features of both users and subreddits are obtained by aggregating the features of their associated posts.

For Cora, ACM, BlogCatalog, Flickr, and ogbn-Arxiv, we introduce the anomaly infection details because they lack ground-truth anomalies. Contextual anomalies are defined as nodes whose attribute representations deviate significantly from those of their neighboring nodes. To inject contextual anomalies, we randomly select m nodes as target nodes. For each target node i, we randomly sample n nodes from the graph as candidate nodes. Among these candidates, we identify the node j that maximizes the Euclidean distance in the feature space, i.e.,

$$j = \arg\max_j \|\mathbf{x}_i - \mathbf{x}_j\|_2 .$$

We then replace the original feature vector $\mathbf{x}_i$ with $\mathbf{x}_j$, thereby transforming node i into a contextual anomaly. Through this process, a total of m contextual anomalies are injected. We set $n = 50$. Structural anomalies refer to nodes that exhibit abnormal connectivity patterns, characterized by unusually dense connections compared to regular nodes. To generate structural anomalies, we randomly select m nodes and fully connect them to form a clique. All nodes within the clique are regarded as structural anomalies. This process is repeated n times, resulting in the injection of $m \times n$ structural anomalies in total. We set $m = 15$.

Since these datasets were not originally intended for federated learning, we split the nodes to generate local datasets for distributed clients. Following (Ghosh et al., 2018), we employ several heuristics to speed up the computation of the different steps of the Louvain algorithm. The Louvain algorithm consists of two main phases, including local modularity optimization and community aggregation, and works by optimizing a metric called modularity, which measures how well a network is partitioned into communities.

Table 8: The details of datasets

Dataset	Nodes	Edges	Anomaly Rate(%)	Attributes
Cora	2708	5429	4.5	1433
ACM	16484	164350	3.62	8337
BlogCatalog	5196	171743	3.6	8189
Flickr	7,575	239738	5.94	500
ogbn-Arxiv	169343	1166243	3.5	128
T-Finance	39357	21222543	4.58	10
Amazon	10224	351216	6.78	25
Reddit	10984	168016	3.33	64

E.2 BASELINES

The details of baselines are as follows:

- UMGAD leverages attribute-level and subgraph-level augmented-view graphs for reconstruction and implements contrastive learning between original and augmented views to improve the model's ability to capture anomalies.
- G3AD introduces double networks based on correlation information to prevent the GNNs from inconsistent information encoding and develops an adaptive caching module to protect the reconstruction from anomalies.
- CARE captures both local and global node affinities by augmenting the graph's adjacency matrix with the pseudo-label. Then it introduces a similarity-guided loss to mitigate potential biases from the pseudo-label.
- FAGAD adaptively integrates signals across multiple frequency bands based on full-pass signals and leverages self-supervised learning to learn discriminative representations.

- PFGA utilizes generative models as bridges to enable information sharing among clients, thereby facilitating the collaborative training of graph representation learning models.
- FedC4 employs graph condensation to refine the knowledge of each client and leverage reconstruction to learn the node representations.
- FedPGT employs a transformer to capture long-term dependencies between client nodes, and pools client subgraphs into an averaged node representation for subgraph similarity calculation.
- FedCAD utilizes pseudo-label discovery to determine the anomaly node of the client and employs a local anomaly neighbor embedding aggregation strategy to aggregate the neighbor embeddings of anomaly nodes from other clients.
- FedCLGN generates global negative pairs and employs graph diffusion to enhance the discrepancy between positive and negative pairs.

F Evaluation Metrics

All baselines are implemented using the source code. Experiments are conducted in a Python 3.9.12 environment on a single NVIDIA A40 GPU with 40 GB memory, 40 GB system RAM, and an Intel Xeon(R) Gold 6240 CPU running at 2.60 GHz. The details of the metrics are as follows:

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**: It measures the model's ability to distinguish between positive (anomalous) and negative (normal) samples by computing the area under the ROC curve, which plots the true positive rate against the false positive rate at various threshold settings. A higher AUROC value indicates stronger discriminative capability across different decision thresholds.
- **Area Under the Precision–Recall Curve (AUPRC)**: It focuses more on the precision–recall trade-off, making it particularly informative in imbalanced settings where anomalies are rare. It reflects how well the model maintains high precision (few false alarms) while achieving high recall (detecting most anomalies).

G Ablation

G.1 Illustration

We illustrate the variants in Table 2. UnFedGAD$_{\text{PP}}$ denotes UnFedGAD without cross-client topology suture. UnFedGAD$_{\text{REC}}$ denotes UnFedGAD without reconstruction loss $\mathcal{L}_{\text{rec}}$.

G.2 More Ablation

In this section, we introduce three variants, including UnFedGAD$_g$, UnFedGAD$_l$, and UnFedGAD$_w$, as shown in Table 9. UnFedGAD$_g$ denotes UnFedGAD, which only calculates local consistency; UnFedGAD$_l$ denotes UnFedGAD, which only calculates global consistency; UnFedGAD$_w$ denotes the removal of the warm-up strategy. From the observations, we can see that global and local consistency learning both have a significant impact on the detection performance, demonstrating the effectiveness of both levels. The performance degradation of UnFedGAD$_w$ demonstrates the effectiveness of the warm-up strategy in alleviating the influence of high noise on training in the early training stage.

H Sensitivity of Threshold Across Unseen Datasets

To validate the sensitivity of the threshold K and $\hat{K}$ on unseen datasets, we extend the experiment settings. We utilize Cora as a training dataset and evaluate the performance on other datasets with a fixed threshold to observe the change in performance. As shown in Table 10 and Table 11, for hyperparameter K, we discover that the performance on unseen datasets has significant degradation, which is because the distribution of the training and testing datasets is largely distinct, and the behavior patterns of nodes have no direct transferability. In addition, in the federated settings, nodes

Table 9: The results of the ablation study. Best results are in bold. All results are in %.

Metric	Variants	Dataset							
		Cora	ACM	BlogCatalog	Flickr	ogbn-Arxiv	T-Finance	Amazon	Reddit
AUROC	UnFedGAD$_g$	$62.84_{\pm 2.7}$	$55.31_{\pm 2.9}$	$50.92_{\pm 2.6}$	$48.06_{\pm 2.8}$	$52.37_{\pm 3.0}$	$73.14_{\pm 2.3}$	$62.88_{\pm 2.9}$	$52.95_{\pm 2.6}$
	UnFedGAD$_l$	$65.02_{\pm 2.6}$	$60.47_{\pm 3.4}$	$53.36_{\pm 2.8}$	$52.41_{\pm 2.7}$	$59.92_{\pm 1.8}$	$70.08_{\pm 2.9}$	$59.55_{\pm 2.6}$	$51.12_{\pm 2.8}$
	UnFedGAD$_w$	$68.96_{\pm 2.2}$	$53.97_{\pm 2.1}$	$59.07_{\pm 1.8}$	$53.49_{\pm 1.8}$	$53.25_{\pm 2.8}$	$75.80_{\pm 2.2}$	$70.33_{\pm 1.8}$	$55.86_{\pm 1.7}$
	UnFedGAD	$\mathbf{80.23}_{\pm 1.4}$	$\mathbf{76.55}_{\pm 1.2}$	$\mathbf{71.47}_{\pm 1.5}$	$\mathbf{68.02}_{\pm 1.3}$	$\mathbf{70.50}_{\pm 1.2}$	$\mathbf{81.36}_{\pm 1.5}$	$\mathbf{83.10}_{\pm 1.6}$	$\mathbf{56.20}_{\pm 1.1}$
AUPRC	UnFedGAD$_g$	$30.96_{\pm 2.9}$	$14.32_{\pm 2.6}$	$17.08_{\pm 3.0}$	$10.24_{\pm 2.8}$	$11.42_{\pm 2.9}$	$40.18_{\pm 2.6}$	$45.06_{\pm 3.0}$	$6.12_{\pm 2.9}$
	UnFedGAD$_l$	$32.04_{\pm 2.7}$	$18.46_{\pm 3.0}$	$20.88_{\pm 2.4}$	$14.86_{\pm 2.6}$	$18.74_{\pm 3.1}$	$38.92_{\pm 3.0}$	$43.38_{\pm 2.8}$	$5.08_{\pm 2.5}$
	UnFedGAD$_w$	$35.24_{\pm 1.6}$	$11.63_{\pm 2.1}$	$25.87_{\pm 2.0}$	$14.05_{\pm 3.1}$	$11.07_{\pm 1.5}$	$39.31_{\pm 1.2}$	$52.76_{\pm 1.8}$	$5.81_{\pm 2.2}$
	UnFedGAD	$\mathbf{46.26}_{\pm 1.6}$	$\mathbf{30.59}_{\pm 1.4}$	$\mathbf{38.16}_{\pm 1.8}$	$\mathbf{26.62}_{\pm 1.5}$	$\mathbf{30.62}_{\pm 1.4}$	$\mathbf{50.04}_{\pm 1.9}$	$\mathbf{60.84}_{\pm 2.0}$	$\mathbf{10.80}_{\pm 1.3}$

are split into different clients, which also exacerbates the difficulty of model learning. For K, the optimal range is from 0.2 to 0.3, which indicates that the proportion of abnormal samples in pseudo-labels should ideally not exceed 30%. For $\hat{K}$, different datasets show varying sensitivity. For example, ACM and BlogCatalog prefer a relatively small value of $\hat{K}$, which indicates that they need to absorb more cross-client neighboring information. However, Reddit prefers a relatively large value of $\hat{K}$, suggesting that it only needs small-scale supplementation.

Table 10: The AUROC performance of K. The model is trained on the Cora dataset and tested on other unseen datasets.

Dataset	$K = 0.1$	$K = 0.2$	$K = 0.3$	$K = 0.4$
ACM	$64.28_{\pm 1.2}$	$70.13_{\pm 2.1}$	$68.44_{\pm 1.5}$	$64.50_{\pm 1.9}$
BlogCatalog	$52.36_{\pm 1.8}$	$55.41_{\pm 2.3}$	$54.81_{\pm 1.6}$	$50.16_{\pm 2.0}$
Flickr	$53.55_{\pm 1.4}$	$58.40_{\pm 2.0}$	$55.94_{\pm 1.7}$	$51.69_{\pm 2.2}$
ogbn-Arxiv	$62.11_{\pm 1.5}$	$64.37_{\pm 2.2}$	$61.78_{\pm 1.9}$	$59.06_{\pm 2.4}$
T-Finance	$61.20_{\pm 1.3}$	$67.85_{\pm 2.4}$	$65.19_{\pm 1.8}$	$60.82_{\pm 2.1}$
Amazon	$67.36_{\pm 2.0}$	$72.91_{\pm 2.5}$	$76.19_{\pm 2.8}$	$68.33_{\pm 2.2}$
Reddit	$49.52_{\pm 1.1}$	$51.69_{\pm 1.9}$	$53.14_{\pm 2.0}$	$51.82_{\pm 1.7}$

Table 11: The AUROC performance of $\hat{K}$. The model is trained on the Cora dataset and tested on other unseen datasets.

Dataset	$\hat{K} = 0.2$	$\hat{K} = 0.4$	$\hat{K} = 0.6$	$\hat{K} = 0.8$
ACM	$70.20_{\pm 1.5}$	$73.33_{\pm 2.1}$	$72.34_{\pm 1.9}$	$59.78_{\pm 2.3}$
BlogCatalog	$59.76_{\pm 1.8}$	$59.51_{\pm 2.0}$	$54.76_{\pm 1.6}$	$52.61_{\pm 1.4}$
Flickr	$54.19_{\pm 1.2}$	$60.84_{\pm 2.2}$	$55.60_{\pm 1.7}$	$52.65_{\pm 1.9}$
ogbn-Arxiv	$59.21_{\pm 1.4}$	$63.68_{\pm 2.0}$	$65.76_{\pm 2.4}$	$52.57_{\pm 1.6}$
T-Finance	$58.80_{\pm 1.3}$	$66.65_{\pm 2.3}$	$64.92_{\pm 1.8}$	$59.29_{\pm 2.1}$
Amazon	$72.26_{\pm 1.9}$	$73.39_{\pm 2.2}$	$65.61_{\pm 2.0}$	$64.73_{\pm 1.7}$
Reddit	$42.82_{\pm 1.1}$	$48.94_{\pm 1.8}$	$57.19_{\pm 2.1}$	$58.92_{\pm 1.9}$

I Utility of Personalized Updating Mechanism

I.1 Experiment with Traditional Updating Mechanism

Fig. 4 shows the AUPRC performance on five datasets. From the observation, we can see that UnFedGAD maintains superiority over the traditional updating mechanism. Local exhibits the worst performance because Local forbids interaction between clients, which fails to solve the distribution shift problem caused by the federated setting. Other baselines perform better than Local while underperforming compared to the proposed method. In addition, Fig. 5 shows the performance on the remaining datasets. UnFedGAD significantly outperforms other variants consistently, which comprehensively demonstrates the advancement of UnFedGAD.

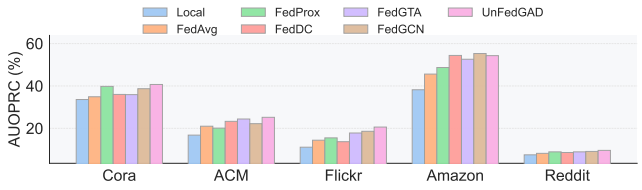

Figure 4: The AUPRC results of the personalized updating mechanism experiments on five datasets.

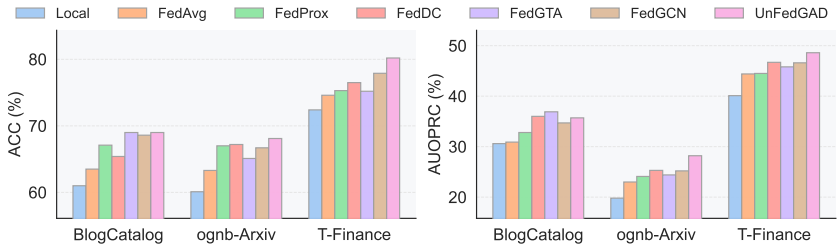

Figure 5: The performance of the personalized updating mechanism experiments on the rest of the datasets.

I.2 The Validation of Components

The details of the variants in Table 3 are as follows. UnFedGAD$_{\text{g(t)}}$ denotes deleting the first term $\Theta_g(t)$ in Eq. (14), UnFedGAD$_{\text{l(t)-g(t)}}$ denotes replacing $\Theta_l(t) - \Theta_g(t)$ in Eq. (14) with $\Theta_l(t)$, UnFedGAD$_{\text{avg}}$ denotes replacing the weights $\bar{w}_m$ with traditional average function, UnFedGAD$_{m'}$ denotes deleting the second term $\Theta_l^{m'}(t)$ in Eq. (16), UnFedGAD$_{mm'}$ denotes deleting the second term $\Theta_l^{m'}(t)$ and the third term $\Theta_l^{m}(t)$ in Eq. (16), and UnFedGAD$_w$ denotes deleting the adaptive weights $\bar{\mathbf{w}}_m$ of $\Theta_g(t+1)$ and $\Theta_l^{m'}(t)$ in Eq. (16).

Table 12: The AUROC of the evaluation of the components in the personalized updating mechanism (Eq. (14) and Eq. (16)) on four datasets. Best results are in bold. All results are in %.

Variants	Dataset			
	ogbn-Arxiv	T-Finance	Amazon	Reddit
UnFedGAD$_{\text{g(t)}}$	$66.84_{\pm 2.6}$	$74.92_{\pm 2.1}$	$77.48_{\pm 2.9}$	$50.92_{\pm 2.4}$
UnFedGAD$_{\text{l(t)-g(t)}}$	$65.12_{\pm 3.0}$	$75.38_{\pm 2.7}$	$76.02_{\pm 2.4}$	$49.36_{\pm 2.1}$
UnFedGAD$_{\text{avg}}$	$67.48_{\pm 2.2}$	$76.06_{\pm 2.5}$	$78.24_{\pm 2.1}$	$51.64_{\pm 1.9}$
UnFedGAD$_{m'}$	$68.92_{\pm 1.9}$	$77.84_{\pm 2.0}$	$79.36_{\pm 2.3}$	$52.18_{\pm 2.2}$
UnFedGAD$_{mm'}$	$68.34_{\pm 2.4}$	$77.26_{\pm 1.8}$	$78.92_{\pm 2.6}$	$51.72_{\pm 2.0}$
UnFedGAD$_w$	$69.08_{\pm 2.1}$	$78.12_{\pm 2.3}$	$79.48_{\pm 2.0}$	$52.64_{\pm 1.8}$
UnFedGAD	$\mathbf{70.50}_{\pm 1.2}$	$\mathbf{81.36}_{\pm 1.5}$	$\mathbf{83.10}_{\pm 1.6}$	$\mathbf{56.20}_{\pm 1.1}$

We further showcase the observations on the remaining four datasets in Table 12. From the observations, the proposed personalized updating mechanism remains superior to variants, which demonstrates the effectiveness of each component in Eq. (14) and Eq. (16). From these experiments, we can conclude that preserving prior knowledge and designing an adaptive fusion mechanism are helpful for high-quality global consensus.

J Utility of GNN Encoders

We introduce four graph encoders, including GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018), GraphSAGE (Hamilton et al., 2017), and GINE (Hu et al., 2020b), to explore the influence of the GNN encoders. As shown in Table 13, in terms of AUROC, GIN achieves the best performance in 5 out of 8 datasets, while it attains the top results in 6 out of 8 cases with respect to AUPRC. In addition, GINE and GraphSAGE comprehensively outperform GCN and GAT. Therefore, it can be

Table 13: The results on different encoder variants. Best results are in bold. All results are in %.

Metric	Encoder	Dataset							
		Cora	ACM	BlogCatalog	Flickr	ogbn-Arxiv	T-Finance	Amazon	Reddit
AUROC	GCN	78.46$_{\pm 1.3}$	69.38$_{\pm 3.1}$	68.12$_{\pm 2.5}$	66.92$_{\pm 2.0}$	66.08$_{\pm 3.1}$	75.36$_{\pm 2.6}$	81.42$_{\pm 1.7}$	53.84$_{\pm 2.5}$
	GAT	77.02$_{\pm 2.5}$	69.96$_{\pm 2.3}$	69.48$_{\pm 2.6}$	65.36$_{\pm 2.2}$	68.42$_{\pm 2.2}$	76.88$_{\pm 1.7}$	80.04$_{\pm 1.4}$	54.26$_{\pm 2.6}$
	GraphSAGE	78.38$_{\pm 1.8}$	70.02$_{\pm 3.2}$	70.42$_{\pm 2.3}$	67.84$_{\pm 1.9}$	**71.48**$_{\pm 2.3}$	78.12$_{\pm 2.3}$	82.26$_{\pm 1.2}$	54.58$_{\pm 2.3}$
	GINE	**81.64**$_{\pm 2.0}$	73.26$_{\pm 1.9}$	71.18$_{\pm 2.1}$	66.02$_{\pm 1.8}$	67.92$_{\pm 2.1}$	79.84$_{\pm 2.2}$	81.52$_{\pm 3.0}$	**56.86**$_{\pm 2.1}$
	GIN	80.23$_{\pm 1.4}$	**76.55**$_{\pm 1.2}$	**71.47**$_{\pm 1.5}$	**68.02**$_{\pm 1.3}$	70.50$_{\pm 1.2}$	**81.36**$_{\pm 1.5}$	**83.10**$_{\pm 1.6}$	56.20$_{\pm 1.1}$
AUPRC	GCN	44.26$_{\pm 2.5}$	27.18$_{\pm 1.2}$	35.84$_{\pm 2.6}$	25.92$_{\pm 2.1}$	24.14$_{\pm 2.5}$	44.48$_{\pm 2.7}$	57.12$_{\pm 2.4}$	6.02$_{\pm 2.6}$
	GAT	43.04$_{\pm 2.4}$	26.92$_{\pm 1.4}$	36.26$_{\pm 2.8}$	22.86$_{\pm 2.3}$	28.86$_{\pm 2.7}$	45.96$_{\pm 2.9}$	56.88$_{\pm 2.6}$	6.74$_{\pm 2.8}$
	GraphSAGE	43.38$_{\pm 2.4}$	28.02$_{\pm 2.1}$	36.48$_{\pm 1.5}$	25.28$_{\pm 3.0}$	**31.92**$_{\pm 2.4}$	48.18$_{\pm 1.6}$	58.02$_{\pm 2.3}$	7.62$_{\pm 2.5}$
	GINE	**46.72**$_{\pm 2.1}$	29.24$_{\pm 2.0}$	37.64$_{\pm 2.3}$	24.42$_{\pm 1.9}$	27.08$_{\pm 2.3}$	49.36$_{\pm 2.5}$	57.46$_{\pm 2.2}$	10.05$_{\pm 1.4}$
	GIN	46.26$_{\pm 1.6}$	**30.59**$_{\pm 1.4}$	**38.16**$_{\pm 1.8}$	**26.62**$_{\pm 1.5}$	30.62$_{\pm 1.4}$	**50.04**$_{\pm 1.9}$	**60.84**$_{\pm 2.0}$	**10.80**$_{\pm 1.3}$

seen that using different GNN encoders can result in slight fluctuations in performance, which can be attributed to their different message-passing mechanisms.

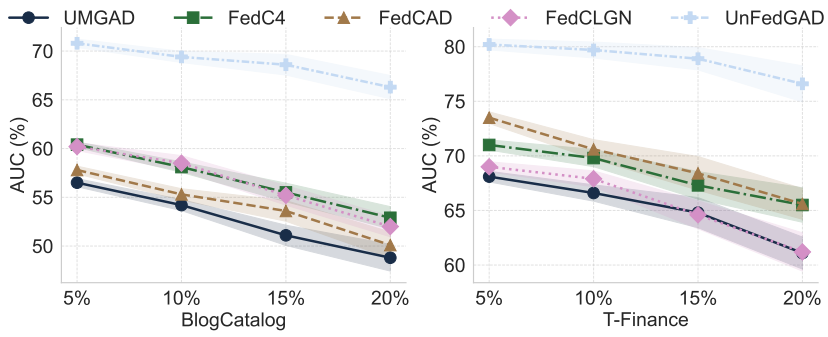

Figure 6: The robustness experiment on BlogCatalog and T-Finance datasets under noise ratios of 5%, 10%, 15%, and 20% with respect to AUROC.

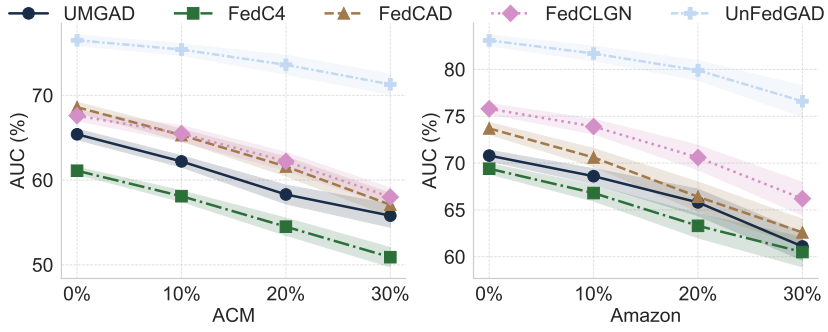

Figure 7: The robustness experiment on ACM and Amazon datasets under noise ratios of 0%, 10%, 20%, and 30% with respect to AUROC.

K Robustness

In this section, we validate the robustness of UnFedGAD by adding standard Gaussian noise to the node features with different proportions. As shown in Fig. 6, the performance on the BlogCatalog and T-Finance datasets demonstrates the superior robustness of UnFedGAD over selected baselines. Specifically, although the performance of UnFedGAD fluctuates with the addition of noise, as the noise rate increases, the performance degradation of the baselines is relatively significant compared to UnFedGAD.

To further demonstrate the noise influence on the performance, Fig. 7 shows the performance degradation under noise ratios of 10%, 20%, and 30%. It can be seen that the performance decline trend is more evident. Moreover, UnFedGAD demonstrates the least performance degradation, indicating superior robustness against noise.

L INSTANCE-LEVEL EVIDENCE

In this section, we report the instance-level evidence that topology suture selects/establishes semantically meaningful cross-client neighbors.

First, we report how many nodes per client receive cross-client semantic connections via topology suture on these datasets. The experimental results in Table 14 provide direct instance-level evidence of how many nodes are influenced by topology suture. Specifically, for each client, more than half of the nodes receive semantic supplementation, establishing connections with nodes across clients, indicating that this method can effectively alleviate the problem of missing semantics caused by sparsity.

Table 14: The instance-level evidence of topology suture on Cora and Flickr datasets.

client	1	2	3	4	5	6	7	8	9	10
Total # of Nodes (Cora)	270	270	270	270	270	270	270	270	270	278
The sutured # of Nodes (Cora)	242	251	214	228	227	209	180	254	192	248
Total # of Nodes (Flickr)	757	757	757	757	757	757	757	757	757	762
The sutured # of Nodes (Flickr)	543	486	502	387	410	465	356	525	467	369

Second, we visualize the distribution of the number of cross-client semantic neighbors per node for clients on the Cora dataset. From the observations in Table 15, we can see that on both datasets, most nodes in each client have extensive semantic supplementation. Generally speaking, most nodes receive between 1 and 20 cross-client sutured connections. Some nodes don't receive any supplementary information, which may be because they show distinct discrepancies from most nodes.

Table 15: The number of cross-client semantic neighbors per node on Cora and Flickr datasets.

client	1	2	3	4	5	6	7	8	9	10
0	28	19	56	42	43	61	90	16	78	30
(0, 10]	142	134	156	103	122	97	88	115	105	97
(10, 20]	83	114	43	115	95	103	73	107	80	140
¿20	17	3	15	10	10	9	19	32	7	11

Table 16: The noise propagation experiment on the Cora and ACM datasets.

Cora					ACM				
Epoch	1	20	50	100	Epoch	1	20	50	100
Homo	216	267	223	356	Homo	2810	4976	3101	815
Heter	117	129	73	66	Heter	1886	2153	815	438
After	945	1008	908	1034	After	10563	12996	9783	9642
Before	612	612	612	612	Before	5867	5867	5867	5867
Noise rate	0.35	0.32	0.24	0.15	Noise rate	0.40	0.30	0.20	0.11

M NOISE PROPAGATION EXPERIMENT

We record the connection quality introduced by the topology suture at different training epochs on Cora and ACM. In the following tables, Homo and Heter denote the number of sutured connections where both endpoints belong to the same or different categories, respectively. Before and After denote the total number of connections before and after the topology suture.

From the results in Table 16, topology suture introduces homophilic semantics for feature convergence, but also brings heterophilic noise. Nevertheless, the Noise rate drops during training (0.35 $\rightarrow$ 0.15 on Cora, 0.40 $\rightarrow$ 0.11 on ACM), confirming that the affinity matrix quality improves as training proceeds and higher-quality semantics are obtained in later stages. We acknowledge that noise cannot be completely avoided during suturing.

N ALGORITHM

The pseudocode is shown in **??** 1.

Algorithm 1: The pseudocode of UnFedGAD.

Input: Client set $\mathcal{C} = \{C_1, \cdots, C_M\}$; Client set $\mathcal{C} = \{C_1, \cdots, C_M\}$; Client epochs T_c; Communication epochs T_s
Output: Cluster results;
Initialize parameters $\mathbf{W}_s$ of the server.
for $t_s = 1$ **to** T_s **do**
  for C_m **in** $\mathcal{C}$ **do**
    if $t_s = 1$ **then**
      Directly download parameters from the server.
    else
      Download parameters from the server via Eq. (16).
    end if
    for $t_c = 1$ **to** T_c **do**
      Obtain the latent $\mathbf{H}^m$ via Eq. (1).
      Implement privacy protection and generate encrypted embeddings $\hat{\mathbf{H}}^m$ via Eq. (2).
      Calculate loss of privacy protection via Eq. (3).
      Calculate prototypes and discover the nearest neighboring client via Eq. (4)-Eq. (6).
      Generate new input features via Eq. (7).
      Implement local-global consistency estimation via Eq. (8)-Eq. (12).
    end for
  end for
  Update the local parameters using the personalized updating mechanism by Eq. (13)-Eq. (14).
end for