FedSDR: Federated Graph Learning with Structural Noise Detection and Reconstruction

Jiaqi Liu† Zihan Tan† Guancheng Wan Wenke Huang* He Li Mang Ye*
National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Wuhan University, Wuhan, China.
{jiaqi.liu,zihantan,guanchengwan,wenkehuang,lihe404,yemang}@whu.edu.cn

Abstract

Federated Graph Learning (FGL) has emerged as a principled framework for decentralized training of Graph Neural Networks (GNNs) while preserving data privacy. In subgraph-FL scenarios, however, structural noise arising from data collection and storage can damage the GNN message-passing scheme of clients, leading to conflicts in collaboration. Existing approaches exhibit two critical limitations: 1) Globally, they fail to identify corrupted clients, causing destructive knowledge inconsistencies. 2) Locally, the global GNN performs poorly on these clients due to structural noise, limiting their ability to benefit from federated collaboration. To address these challenges, we propose ***FedSDR****, a spectra-based FGL framework against high-structural-noise scenarios. Specifically, Structural Noise-Aware Aggregation (SNAA) introduces a structural fidelity evaluation metric to detect corrupted clients and reduce their contributions, thereby mitigating the impact of noise on the global GNN. Furthermore, Robust Local Structure Reconstruction (RLSR) leverages the knowledge from the healthy global model to repair locally corrupted graph structures. Extensive experiments demonstrate that* ***FedSDR*** *outperforms state-of-the-art methods across various scenarios under structural noise. The code is available at* https://github.com/Subtleazure/FedSDR.

1. Introduction

Graph Neural Networks (GNNs) [37, 41, 44, 67] have emerged as a powerful framework for learning from graph-structured data, leveraging message-passing mechanisms to capture complex relational patterns. However, traditional GNNs training paradigms rely on centralized access to the entire graph [55, 56], which becomes impractical in the real world due to privacy considerations [63]. This limitation is particularly acute in domains such as healthcare [35], finance [42], and social networks [63]. To address these challenges, recent advances have integrated Federated Learning (FL) [29, 39] with GNNs, giving rise to Federated Graph Learning (FGL) [48, 50, 64]. FGL extends FL principles to decentralized graph data, enabling clients to collaboratively train GNNs while preserving their privacy.

However, the increasing scale of distributed systems introduces vulnerabilities, as transmission errors, system malfunctions, or malicious activities may lead some clients to

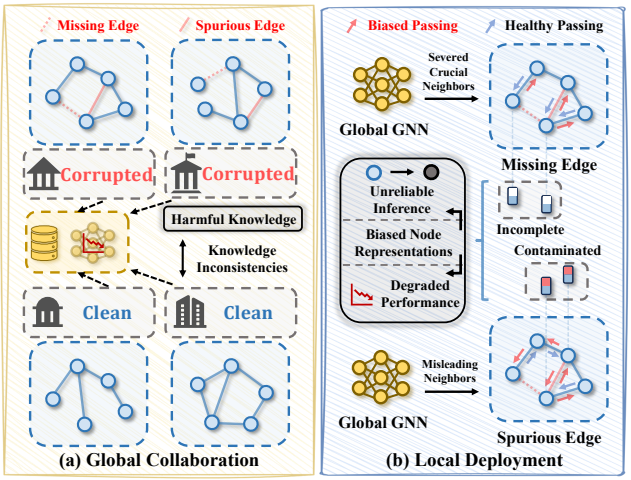
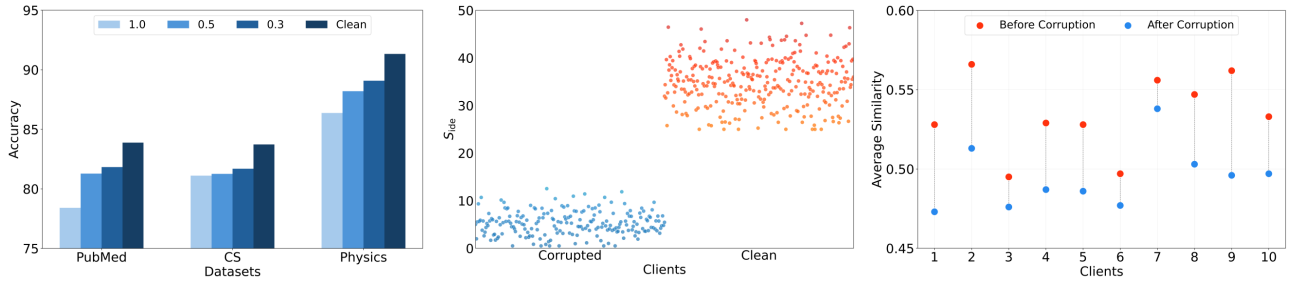


Figure 1. Problem illustration. The structural integrity of client graphs is fundamental to the efficacy of both local and global message-passing schemes. Structural noise in local graphs critically disrupts this integrity, creating two primary challenges under high-structural-noise scenarios. (a) Globally, structural noise damages the message-passing scheme of clients, introducing **harmful knowledge** that causes **knowledge inconsistencies** and undermines collaborative learning. (b) Locally, such noise **disrupts** message-passing schemes of the global model, resulting in **biased node representations**, yielding unreliable inference and degraded model performance on corrupted clients.

*Corresponding author. † Equal contribution.



(a) Performance Degradation **(b) Fidelity Metric Distribution** **(c) Corruption Impact on Similarity**

Figure 2. Phenomenon Demonstration. (a) The test accuracy comparison between models trained on corrupted and clean graphs under FedAvg, with a corruption ratio of 1 and noise extents of 0.3, 0.5 and 1 (implementation in Sec. 5.1). It reveals corrupted clients impose **harmful knowledge**, undermining the model performance. (b) The fidelity metric for corrupted clients is significantly **lower** than for clean ones, confirming its utility for reliable noise detection. (c) Under SNAA strategy, the average feature similarity of local subgraphs is significantly **lower** after the introduction of structural corruption compared to their clean states. It demonstrates that RLSR is well-founded.

submit incorrect updates [2, 22, 38]. Conventional algorithms such as FedAvg and FedSGD can fail even with a single corrupted client, motivating the development of robust strategies. Existing robust FGL methods primarily focus on adversarial scenarios [10, 53] or natural topological heterogeneity [14, 33]. However, real-world corruption often occurs randomly, creating a mismatch: these methods overestimate adversarial threats or misinterpret random structural noise as mere heterogeneity, leading to unnecessary performance compromises and misidentification. Crucially, such non-malicious **structural noise**: **large-scale, random topology corruption** including **missing** and **spurious edges** from practical data imperfections, often disguises itself as benign heterogeneity, evading detection by both adversarial-defense and heterogeneity-handling approaches. Moreover, this noise is prevalent in real-world graph applications. For instance, social networks like Facebook or LinkedIn frequently contain substantial inauthentic follower relationships introduced by bot accounts or fake profiles [23]. As shown in Fig. 2(a), such noise severely degrades model performance, highlighting two critical deficiencies in practical deployment as detailed below.

For global collaboration, structural noise can damage the GNN message-passing scheme of clients. Therefore, structurally corrupted clients often impose harmful knowledge on the global GNN, introducing destructive global knowledge inconsistencies. Existing methods assume an originally intact graph topology, rarely addressing structural noise. Specifically, [62] only generates missing neighbors to handle cross-subgraph link sparsity, and [15] indirectly measures neighbor information deviation through statistical indicators. Neither method effectively identifies structural noise, leading to performance degradation in structural noise scenarios. The key to addressing structural noise is identifying structurally corrupted clients and reducing their impact. This observation raises a key question: 1) *How can we **detect** and **mitigate the adverse impact of corrupted clients** under high-structural-noise scenarios?*

For local deployment, global model message-passing schemes on corrupted clients are fundamentally undermined by structural noise. Missing edges sever information pathways, preventing nodes from accessing signals from crucial neighbors and resulting in incomplete or inaccurate embeddings. Conversely, spurious edges introduce noisy connections, leading nodes to aggregate information from irrelevant or misleading neighbors, which contaminates their embeddings. Therefore, both types of edge corruption lead the global GNN to compute biased node representations on corrupted graphs, inevitably yielding unreliable inference and degraded model performance. Consequently, it denies corrupted clients any benefits of collaborative learning. This analysis triggers the following question: 2) *How to design an effective method to **repair corrupted graph structure** without compromising valuable knowledge during aggregation?*

To address the first challenge, motivated by the relationship between graph structure and spectral properties [7, 26, 48], we hypothesize that structural noise will manifest as detectable anomalies in the spectral domain. Moreover, we develop an innovative spectral fidelity evaluation metric accordingly (Sec. 3.1). To validate this, we first conduct experiments revealing that structurally corrupted clients exhibit a significantly lower fidelity metric than clean ones (Fig. 2(b)). Building on this insight, we introduce spectra-guided **Structural Noise-Aware Aggregation** (SNAA). This principled method detects structurally corrupted clients and reduces their contributions through a weighting mechanism based on their extent of corruption. By dynamically redistributing client contributions, SNAA effectively mitigates the adverse impact of corrupted clients, fostering resilient and adaptive collaborative learning in the presence of structural noise.

To tackle the second issue, we develop a graph repair strategy. Since SNAA inherently establishes a reliable global

message-passing scheme, it synthesizes healthy global knowledge that suggests consensus among clean clients. This consensus is strongly reflected in the feature similarity, which reflects semantic validity [21]. Therefore, clients with structural noise exhibit significantly lower similarity against this knowledge (Fig. 2(c)). Motivated by this, we propose **Robust Local Structure Reconstruction** (RLSR), which leverages the knowledge of the healthy global model to guide local graph repair. In contrast to solely removing the spurious edges, our approach selectively reconnects potentially missing edges based on similarity alignment with the global model, while maintaining existing valid local structure patterns. Moreover, it corrects the intrinsic spectral properties of the graph (Fig. 3), which are critical for effective GNN message-passing schemes [3, 16, 45]. Therefore, the repaired structure remains consistent with collaboratively learned knowledge and enhances robust GNN training. This reconstruction in turn improves the noise detection effectiveness of SNAA. In synergy, this framework enables the model to learn valid structural distributions while achieving fine-grained decoupling of corrupted and clean clients. Combining both strategies, we propose **FedSDR**, a novel framework **Fed**erated Graph Learning with **S**tructural noise **D**etection and **R**econstruction. Our principal contributions are summarized as follows.

- We are the first to reveal that structurally corrupted clients manifest as spectral outliers, and that corrupted connections introduce observable discrepancies in feature similarities between the local and global models.
- We propose FedSDR, an innovative framework that detects structurally corrupted clients from a spectral perspective and mitigates their adverse impact through reweighting and structure reconstruction.
- We conducted extensive experiments in high-structural-noise settings to demonstrate that FedSDR outperforms state-of-the-art methods, achieving superior robustness and accuracy.

2. Related Work

2.1. Graph Neural Networks

Graph Neural Networks (GNNs) [57, 61, 65] have emerged as a powerful framework for learning representations of graph-structured data, broadly categorized into spectral and spatial approaches. Spectral GNNs [4, 8] design filters in the spectral domain, with early methods such as ChebNet [11] employing fixed polynomial approximations and more recent advances, for instance, BernNet [18] and JacobiConv [54], introducing scalable spectral filters for enhanced flexibility. In contrast, spatial GNNs [9, 59] operate via direct neighborhood aggregation, with influential examples including GIN [58], GAT [51], and GraphSAGE [17]. While spatial methods excel in scalability and inductive learning, they can face challenges in handling structural perturbation and long-range dependencies, issues that spectral methods may mitigate through global frequency-aware filtering. This motivates our work to integrate spectral robustness into federated GNNs, bridging a gap in the existing literature where spectral analysis remains underexplored in decentralized settings.

2.2. Robust Federated Learning

Robust Federated Learning aims to mitigate challenges such as data heterogeneity and adversarial attacks in distributed collaborative training. In heterogeneous scenarios, [12] tackles label noise with a noise-tolerant loss, whereas [60] mitigates feature and structural heterogeneity through inter-intra graph disentanglement. To address byzantine and backdoor attacks, [66] proposes a high-dimensional robust aggregation method with near-optimal statistical guarantees, while [53] leverages topological graph energy to decouple benign and malicious clients. However, existing methods assume originally intact graph topology or natural non-IID data distribution, overlooking a more **disruptive** and **prevalent** form of corruption in real-world deployment: **structural noise**. This problem setting, characterized by non-malicious but extensive edge corruption, has not been the focal point of existing Robust FL literature. Our work explicitly introduces a **spectra-based fidelity metric** that directly assesses graph structural integrity, enabling reliable detection of structural noise even under natural non-IID settings (e.g. those induced by Louvain partitioning) and heterogeneous distributions, as validated in Tab. 1 on datasets including Roman-empire and ogbn-mag.

2.3. Federated Graph Learning

Federated Graph Learning (FGL) enables collaborative training of GNNs across decentralized clients while preserving graph data privacy [36, 49]. Current research is broadly categorized into inter-graph and intra-graph learning. In inter-graph FGL, clients train GNNs on distinct local graphs to achieve globally generalizable models or enhance local performance [47]. Intra-graph FGL focuses on tasks like node classification [34, 50], link prediction [28], and community detection within shared or partitioned graphs [1, 27]. In addition, several methods have been proposed to tackle topological challenges in FGL. For instance, FedATH [14] alleviates topology heterogeneity by extracting causal subgraphs to enhance model generalization, and AdaFGL [33] handles topological variations through adaptive personalized propagation. However, these approaches, designed primarily for data heterogeneity, remain ineffective against random structural corruption, which masquerades as benign heterogeneity, thereby evading conventional detection mechanisms. In contrast, this paper introduces a **novel spectral perspective** to detect and mitigate the ad-

verse impact of intra-graph structural noise. This represents a fundamental shift from merely handling data distribution shifts to diagnosing and healing structural damage, establishing a new direction for robust FGL.

3. Preliminaries

3.1. Structural Fidelity Evaluation Metric

Consider the undirected graph represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the collection of nodes with a total of $|\mathcal{V}| = N$ vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines the set of edges that connect pairs of nodes. The graph corresponds to an adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, where $\mathbf{A}_{uv} = 1$ represents the existing edge $e_{uv} \in \mathcal{E}$ and $\mathbf{A}_{uv} = 0$ otherwise. The degree matrix is constructed as $\mathbf{D} = \text{diag}(d_1, ..., d_N)$, where each $d_i = \sum_{j=1}^{N} \mathbf{A}_{ij}$ represents the degree of node $i \in \mathcal{V}$, with the assertion $d_i > 0$. Then the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Our structural fidelity evaluation metric S_{ide} is defined as:

$$S_{\text{ide}} = \frac{\mathbf{D}^T\mathbf{L}\mathbf{D}}{\mathbf{D}^T\mathbf{D}}, \tag{1}$$

where $\mathbf{D}^T\mathbf{L}\mathbf{D}$ quantifies the connectivity fidelity, while $\mathbf{D}^T\mathbf{D}$ provides normalization across varying graph scales. The efficacy of S_{ide} derives from spectral graph theory, as the eigenvalues of the Laplacian encode graph topology [13, 52]. Specifically, this fraction denotes the ratio of the sum of all elements in the respective matrices. Therefore, the numerator evaluates to the Laplacian quadratic form of the degree vector $\mathbf{d} = [d_1, ..., d_N]^T$: $\mathbf{d}^T\mathbf{L}\mathbf{d} = \sum_{(u,v)\in\mathcal{E}}(d_u - d_v)^2$. This yields $S_{\text{ide}} \propto \sum_{(u,v)\in\mathcal{E}}(d_u - d_v)^2$, which quantifies the total degree discrepancy of connected nodes. Real-world graphs exhibit disassortative mixing (i.e., hubs connecting to leaves), maximizing $(d_{hub} - d_{leaf})^2$ and yielding high S_{ide}. Conversely, random structural noise pushes the topology towards an Erdős-Rényi state, where degrees concentrate around the mean. This randomization homogenizes the degree distribution, significantly reducing the expected degree difference $\mathbb{E}[(d_u - d_v)^2]$ and causing the metric to decay, validating our metric for reliable noise detection.

3.2. Federated Learning

Federated learning establishes a decentralized optimization paradigm where K distributed clients collaboratively train machine learning models while maintaining data locality. In federated optimization, we consider a global model with parameters $\theta \in \mathbb{R}^d$, where d denotes the parameter dimension. The collaborative objective minimizes:

$$\min_{\theta} \mathcal{F}(\theta) = \min_{\theta} \sum_{k=1}^{K} w_k \mathcal{L}_k(\theta), \tag{2}$$

where $\mathcal{F}(\theta)$ represents the global optimization objective, measuring the average loss across all clients. $\mathcal{L}_k(\theta) = \mathbb{E}_{(\mathbf{Z},y)\sim\mathcal{P}_k}[\ell(\mathbf{Z}, y)]$ denotes the local expected risk for client k with data distribution $\mathcal{P}_k$. $\ell(\cdot)$ is the loss function evaluating the discrepancy between the model prediction and the true label $(\mathbf{Z}, y)$, and the participation weight $w_k \geq 0$ satisfies $\sum_k w_k = 1$. The training process alternates between local computation and global synchronization. During each communication round t, clients receive the global model θ^t and perform local updates:

$$\theta_k^{t+1} \leftarrow \theta_k^t - \eta\nabla_\theta\mathcal{L}_k(\theta_k^t), \tag{3}$$

where η is the learning rate and $\nabla_\theta\mathcal{L}_k$ denotes the local gradient. The server then aggregates these updates through weighted averaging $\theta^{t+1} = \sum_{k=1}^{K} w_k\theta_k^{t+1}$.

4. Methodology

4.1. Structural Noise-Aware Aggregation

Motivation. The effectiveness of FGL hinges on the ability to collaboratively train models across clients with diverse graph structures. However, structural noise in local graphs can significantly disrupt this process, introducing harmful inconsistencies in the learned representations. To address this challenge, we propose Structural Noise-Aware Aggregation (SNAA). Our method enables us to detect structurally corrupted clients and dynamically adjust their contributions to mitigate their adverse impact. Our approach is grounded in spectral graph theory (Sec. 3.1), leveraging the observation in Fig. 2(b) that structural noise manifests as deviations in the spectral properties of graphs. Details of SNAA are presented in Fig. 3.

Structural Noise Detection. The structural noise detection framework in SNAA stems from spectral graph analysis, where the Laplacian quadratic form $\mathbf{D}^T\mathbf{L}\mathbf{D}$ encodes critical structural properties of a graph, and its spectrum reflects the connectivity patterns and noise levels. Building on the spectral graph formula established in Eq. (1), we develop a rigorous structural fidelity evaluation metric. For client k with Laplacian $\mathbf{L}^k = \mathbf{D}^k - \mathbf{A}^k$, the structural fidelity metric S_{ide}^k is computed locally to capture the extent of structural noise without exposing the raw graph structure:

$$S_{\text{ide}}^k = \frac{\langle \mathbf{D}^{k^T}\mathbf{L}^k\mathbf{D}^k, \mathbf{1}\rangle_F}{\langle \mathbf{D}^{k^T}\mathbf{D}^k, \mathbf{1}\rangle_F} = \frac{\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}[\mathbf{D}^{k^T}\mathbf{L}^k\mathbf{D}^k]_{ij}}{\sum_{i=1}^{N_k}\sum_{j=1}^{N_k}[\mathbf{D}^{k^T}\mathbf{D}^k]_{ij}}, \tag{4}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product, $\mathbf{1}$ is the all-ones matrix, and N_k accounts for the local node counts.

Contribution Redistribution. Building upon these noise assessments, we develop a principled contribution redistribution scheme for robust federated aggregation. The trustworthiness of each client k is quantified through its structural fidelity metric S_{ide}^k, with lower values indicating worse preservation of authentic structural relationships. To mitigate noise-induced biases during model aggregation, we

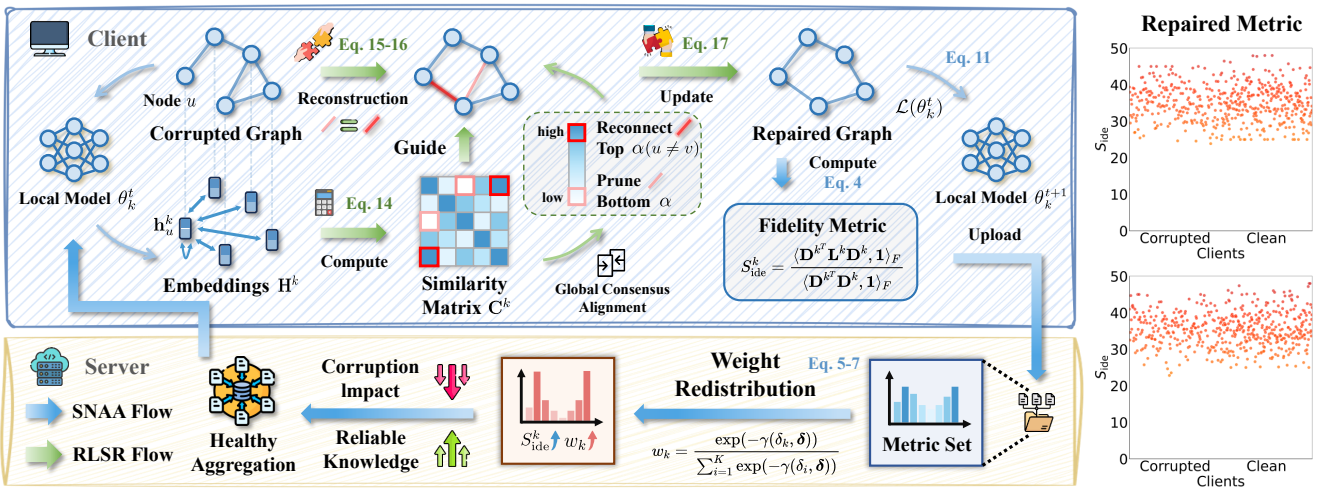


Figure 3. Architecture illustration of FedSDR. We used blue and green arrows to represent the two components of our method, SNAA and RLSR, respectively. In SNAA, each client computes the structural **fidelity metric** S_{ide}^k locally, and the global model dynamically **reweights** their contributions based on this metric, thereby mitigating the influence of corrupted clients. In RLSR, we prune potentially spurious edges with low feature similarity and reconnect an equivalent number of potentially missing edges with high similarity to **align** the local graph structure with the global consensus. The right scatter plots display fidelity metric distribution, showing that RLSR effectively **restores spectral properties**, which are critical for effective message-passing schemes.

first compute the structural noise bias δ_k for each client, measuring its deviation from the global mean noise level:

$$\delta_k = S_{\text{ide}}^k - \frac{\sum_{i=1}^K N_i S_{\text{ide}}^i}{\sum_{j=1}^K N_j}. \tag{5}$$

This weighted formulation ensures fair comparison across clients with varying dataset sizes. A lower δ_k value indicates higher structural noise and thus lower reliability. The resulting bias terms are then normalized through min-max scaling to maintain consistent value ranges:

$$\gamma(\delta_k, \boldsymbol{\delta}) = -\frac{\delta_k \cdot \min(\boldsymbol{\delta})}{\max(\boldsymbol{\delta}) \cdot \min(\boldsymbol{\delta})}, \tag{6}$$

where $\boldsymbol{\delta}$ represents the complete set of client noise biases. The aggregation weight w_k for client k is obtained by applying the negative exponential function to the normalized bias, emphasizing clients with higher S_{ide}^k:

$$w_k = \frac{\exp(-\gamma(\delta_k, \boldsymbol{\delta}))}{\sum_{i=1}^K \exp(-\gamma(\delta_i, \boldsymbol{\delta}))}. \tag{7}$$

Through the exponential transformation, we assign higher weights to clients with cleaner structural patterns and smoothly alleviate the impact of noisier clients. Therefore, this weighting scheme ensures the global model prioritizes reliable knowledge from clean graphs, which is essential for consistent message-passing dynamics across clients.

Reliability-Aware Weighting Aggregation. We design a localized training mechanism to integrate local insights and global knowledge harmoniously. Each client iteratively refines its model using private data, preserving domain-specific expertise while mitigating catastrophic forgetting. The local training procedure generating θ_k^{t+1} ensures compatibility with our noise-aware aggregation. Given node features $\mathbf{X}^k \in \mathbb{R}^{N_k \times f}$ with N_k nodes and f-dimensional features, $\mathcal{E}_k$ the set of edges in the graph of the client k, the forward pass yields node embeddings and logits:

$$\mathbf{Z}^k = f_{\theta_k}(\mathbf{X}^k, \mathcal{E}_k), \tag{8}$$

where $\mathbf{Z}^k \in \mathbb{R}^{N_k \times C}$ denotes the class logits for C classes. Here, $f_{\theta_k}(\cdot)$ represents the local model inference. Let θ^t be the global model parameters at round t, and θ_k^t denote the local parameters for client k. For each node v in the training set, $\mathbf{Z}_v^k \in \mathbb{R}^C$ represents the logits and $y_v^k \in \{1, ..., C\}$ is the true label. The training loss combines cross-entropy, label smoothing and model regularization:

$$\begin{aligned}\mathcal{L}(\theta_k^t) = &\frac{1}{|\mathcal{M}_{\text{train}}|} \sum_{v \in \mathcal{M}_{\text{train}}} \sum_{y_v^k=1}^{C} \ell(\mathbf{Z}_v^k, y_v^k) \\ &+ \frac{\epsilon}{2}\|\mathbf{Z}\|_2^2 + \lambda\|\theta_k^t - \theta^t\|_2^2,\end{aligned} \tag{9}$$

where $\mathcal{M}_{\text{train}}$ is the training node mask, and λ controls the regularization strength. The cross-entropy term $\ell(\cdot)$ follows the precise implementation:

$$\ell(\mathbf{Z}, y) = -\log\left(\frac{\exp(\mathbf{Z}[y])}{\sum_{c=1}^C \exp(\mathbf{Z}[c])}\right), \tag{10}$$

where ϵ controls label smoothing intensity. $\mathbf{Z}[y]$ and $\mathbf{Z}[c]$ index are the y-th and c-th elements of $\mathbf{Z}$, respectively. To

guarantee (ϵ, δ)-differential privacy, we employ a Gaussian mechanism. The update rule for client k is given by:

$$\theta_k^{t+1} \leftarrow \theta_k^t - \eta \nabla_{\theta_k^t} \mathcal{L}(\theta_k^t) + \mathcal{N}\left(0, \left(\frac{B\sigma}{|\mathcal{M}_{\text{train}}|}\right)^2 \mathbf{I}\right). \quad (11)$$

Here, η is the learning rate, B is the gradient clipping threshold, and σ is the noise multiplier. At the round $t+1$, utilizing the aggregation weight w_k defined in Eq. (7) for each client, the global model aggregates updates with reliability-aware weighting:

$$\theta^{t+1} = \sum_{k=1}^{K} w_k \theta_k^{t+1}. \quad (12)$$

By dynamically adjusting contributions based on spectral structural noise evaluation metrics, we align the global model with clean graph characteristics, fostering robust collaboration even under significant structural corruption. Subsequently, we successfully achieve robust training while maintaining model quality, efficiency, and privacy guarantees critical for practical deployment.

4.2. Robust Local Structure Reconstruction

Motivation. While SNAA mitigates the global impact of structural noise, global model performance on corrupted clients remains compromised. Since merely reducing the contributions of corrupted clients discards their potentially valuable knowledge, this can still induce biased representations during local message-passing and ultimately degrade the global model performance on such clients. Our solution leverages global robust feature representations to guide local graph repair, simultaneously preserving authentic local patterns and correcting structural noise. Details of RLSR are presented in Fig. 3.

Since SNAA ensures the global model predominantly reflects clean structural patterns, its node embeddings encode discriminative feature relationships that are resilient to local noise. The core insight of RLSR lies in the fact that structurally spurious and missing edges manifest as deviations in feature similarity space when evaluated against the global model. The normalized feature similarity matrix $\mathbf{C}^k \in \mathbb{R}^{N_k \times N_k}$ captures pairwise relationships:

$$\mathbf{C}^k = \text{diag}(\mathbf{H}^k \mathbf{H}^{k^T})^{-\frac{1}{2}} (\mathbf{H}^k \mathbf{H}^{k^T}) \text{diag}(\mathbf{H}^k \mathbf{H}^{k^T})^{-\frac{1}{2}}, \quad (13)$$

where $\mathbf{H}^k = f_\theta(\mathbf{X}^k, \mathcal{E}_k) \in \mathbb{R}^{N_k \times d}$ be node embeddings from the global model f_θ. Each element in Eq. (13) measures the feature similarity between nodes u and v under the global inductive inference:

$$\mathbf{C}_{uv}^k = \frac{\langle \mathbf{h}_u^k, \mathbf{h}_v^k \rangle}{\|\mathbf{h}_u^k\| \cdot \|\mathbf{h}_v^k\|}, \quad (14)$$

where $\mathbf{h}_u^k$ denotes the embedding of node u in client k. Authentic edges in clean graphs align with high feature similarities, whereas structural noise introduces edges with discordantly low similarities and severs those with high similarity. RLSR exploits this discrepancy through two edge refinement processes.

First, to fundamentally eliminate structural noise, we prune spurious edges. For each client k, we compute a pruning threshold τ_p^k as the α-quantile of existing edge similarities, defined formally through the quantile function $Q(\cdot, \alpha)$:

$$\tau_p^k = Q\left(\{\mathbf{C}_{uv}^k \mid \mathbf{A}_{uv}^k = 1\}, \alpha\right), \quad (15)$$

where $Q(S, \alpha)$ denotes that there are α proportion of elements in S whose values are less than or equal to it, i.e., α proportion of connected edges have similarities below τ_p^k. Edges satisfying $\mathbf{C}_{uv}^k < \tau_p^k$ are identified as spurious connections and removed. This operation eliminates connections that contradict the healthy knowledge of the global model, preserving only those consistent with the consensus of global features.

Second, we perform feature-aware edge reconnection. While the removal of spurious edges is necessary, it risks inducing excessive graph sparsification, which can disrupt the original message-passing dynamics. To compensate for the pruned connections, our subsequent reconnection step establishes new links between nodes with high feature similarity that are likely missing from the corrupted graph structure. Candidate edges are sampled from originally non-adjacent node pairs with similarities exceeding a reconnection threshold τ_r^k, defined as:

$$\tau_r^k = Q\left(\{\mathbf{C}_{uv}^k \mid \mathbf{A}_{uv}^k = 0, u \neq v\}, 1 - \frac{2\alpha|\mathcal{E}^k|}{N_k(N_k-1) - 2|\mathcal{E}^k|}\right), \quad (16)$$

where τ_r^k identifies the top $\alpha|\mathcal{E}^k|$ highest-similarity non-edge, matching the count of pruned edges. New edges are established for pairs (u, v) satisfying $\mathbf{C}_{uv}^k \geq \tau_r^k$, while excluding self-loop reconnection to improve the message-passing scheme. Therefore, reconnected edges exhibit strong feature alignment with the global consensus. The joint pruning-reconnection operation in Eq. (15) and Eq. (16) yields a repaired adjacency matrix $\tilde{\mathbf{A}}^k$:

$$\tilde{\mathbf{A}}_{uv}^k = \begin{cases} 0 & \text{if } \mathbf{A}_{uv}^k = 1 \;\&\; \mathbf{C}_{uv}^k < \tau_p^k, \\ 1 & \text{if } \mathbf{A}_{uv}^k = 0 \;\&\; u \neq v \;\&\; \mathbf{C}_{uv}^k \geq \tau_r^k, \\ \mathbf{A}_{uv}^k & \text{otherwise.} \end{cases} \quad (17)$$

By replacing spurious edges with feature-consistent connections, we succeed in both mitigating structural noise and preserving client-specific patterns unaffected by corruption. In this way, RLSR fundamentally addresses issues of structural noise, establishing a robust foundation for decentralized GNN training in high-structural-noise environments.

Methods	PubMed	Coauthor-CS	Coauthor-Phy	Actor	Roman-empire	ogbn-mag	ogbn-products
FedAvg [ASTAT17]	78.41 ± 0.36	82.01 ± 0.49	86.37 ± 0.27	31.28 ± 0.77	41.72 ± 0.39	42.74 ± 0.22	72.66 ± 1.53
FedProx [arxiv18]	77.82 ± 0.97	74.94 ± 0.36	86.19 ± 0.20	31.27 ± 0.50	41.76 ± 0.71	41.87 ± 0.34	72.51 ± 0.83
Scaffold [ICML20]	41.76 ± 0.83	43.00 ± 3.23	77.41 ± 1.75	23.52 ± 1.35	18.08 ± 0.33	15.09 ± 3.40	59.07 ± 2.21
MOON [CVPR21]	68.55 ± 2.27	75.91 ± 0.29	88.14 ± 0.16	31.14 ± 0.55	42.51 ± 0.30	42.96 ± 0.84	67.94 ± 0.45
Ditto [ICML21]	76.84 ± 1.02	80.33 ± 0.77	85.49 ± 0.55	30.62 ± 0.29	41.30 ± 0.67	41.23 ± 1.20	71.89 ± 0.95
FedSage+ [NIPS21]	78.29 ± 0.93	80.72 ± 0.51	85.83 ± 1.28	30.42 ± 0.74	41.64 ± 0.57	41.92 ± 0.47	71.98 ± 0.40
FedProto [AAAI22]	75.24 ± 0.17	67.26 ± 0.40	84.42 ± 0.23	22.38 ± 0.33	19.42 ± 0.18	37.16 ± 0.29	66.79 ± 1.59
RHFL [CVPR22]	78.07 ± 0.77	79.76 ± 0.87	86.11 ± 1.05	31.27 ± 0.66	42.33 ± 0.45	42.80 ± 0.41	71.04 ± 0.58
FGSSL [IJCAI23]	77.65 ± 0.41	81.05 ± 0.63	84.78 ± 1.61	31.20 ± 0.64	37.41 ± 0.51	41.23 ± 0.75	69.63 ± 1.13
FED-PUB [ICML23]	75.28 ± 0.64	79.34 ± 0.45	85.55 ± 0.87	30.59 ± 0.28	42.12 ± 0.65	40.15 ± 0.56	70.06 ± 0.99
FedTAD [IJCAI24]	77.92 ± 1.02	68.90 ± 0.93	OOM	31.13 ± 0.74	41.29 ± 0.40	OOM	OOM
AdaFGL [ICDE24]	76.63 ± 0.88	79.50 ± 0.56	87.19 ± 0.44	30.90 ± 0.62	41.82 ± 0.57	41.22 ± 0.34	69.68 ± 1.05
FedGTA [VLDB24]	78.04 ± 0.58	81.24 ± 0.61	85.77 ± 0.16	31.03 ± 0.93	40.65 ± 0.21	42.52 ± 0.33	71.52 ± 0.78
FedSSP [NIPS24]	77.46 ± 0.43	79.67 ± 0.23	86.12 ± 0.74	30.52 ± 0.70	39.45 ± 0.62	40.48 ± 0.63	69.36 ± 0.85
FedTGE [ICLR25]	76.35 ± 0.98	77.54 ± 0.59	85.36 ± 0.63	30.25 ± 0.82	39.43 ± 0.78	40.65 ± 0.57	68.11 ± 0.34
FedATH [ICML25]	76.92 ± 0.71	78.95 ± 0.55	86.39 ± 0.48	30.37 ± 0.46	40.02 ± 0.33	41.35 ± 0.18	69.15 ± 0.54
FedIIH [AAAI25]	78.34 ± 0.69	79.34 ± 0.62	87.03 ± 0.80	30.73 ± 0.65	39.84 ± 0.39	42.16 ± 0.47	71.70 ± 0.60
FedSDR (ours)	**82.57 ± 0.64**	**82.29 ± 0.32**	**89.37 ± 0.24**	**32.36 ± 0.59**	**48.33 ± 0.28**	**47.41 ± 0.42**	**78.57 ± 0.28**

Table 1. Comparison with the state-of-the-art methods on non-IID data. The best and second results are highlighted with bold and underline, respectively. Please see additional results in Appendix D.

5. Experiments

5.1. Experimental Setup

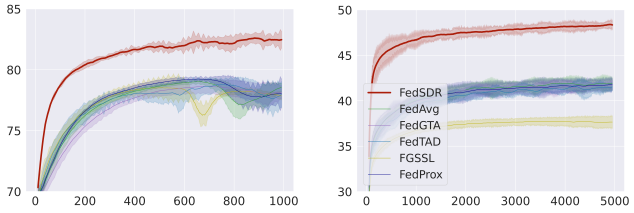
Datasets. We conducted experiments under various structural noise scenarios to validate the superiority of our proposed FedSDR. The datasets with different scales include homophilic datasets PubMed [43], Coauthor-CS, Coauthor-Physics [44], ogbn-products [19] and heterophilic datasets Actor [40], Roman-empire [5], ogbn-mag [19]. Detailed descriptions of these datasets can be found in Appendix B.

Baselines. We compare FedSDR with several state-of-the-art federated approaches. (1) **FedAvg** [39]; (2) **FedGTA** [34]; (3) **FedProto** [46]; (4) **FedProx** [31]; (5) **FedTAD** [68] applying topology-aware knowledge distillation technology; (6) **FGSSL** [20] which reveals distortion brought by node semantics and graph structure, (7) **MOON** [30], (8) **Scaffold** [24], (9) **Ditto** [32] and (10) **RHFL** [12], two state-of-the-art robust FL methods; (11) **FedSage+** [62] and (12) **FED-PUB** [1], two state-of-the-art subgraph FL baselines; (13) **AdaFGL** [33] tackling topology heterogeneity, (14) **FedSSP** [48] accommodating personalized preference, (15) **FedTGE** [53] handling backdoor attacks, (16) **FedATH** [14] alleviating topology heterogeneity and (17) **FedIIH** [60] addressing inter-intra heterogeneity.

Implementation Details. We use the Louvain algorithm [6] to partition the graph, which is an effective strategy to obtain non-IID data in FGL [20, 62]. For homophilic datasets, we adopt a 20%/40%/40% split for training, validation, and testing, whereas for heterophilic datasets, the split is 50%/25%/25%. The local GNN models are trained using the Adam optimizer [25]. The specific learning rate, the number of communication rounds and clients are detailed in Appendix B. To simulate high-structural-noise scenarios, the corruption ratio (proportion of corrupted clients), the noise extent (proportion of edges randomly added and then deleted in corrupted clients), and the pruning proportion α are set to 1, 0.5, and 0.3, respectively. The results represent the average of 5 runs with different random seeds.

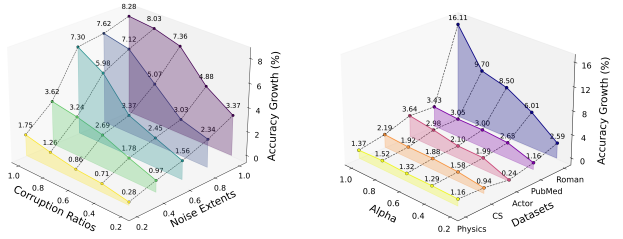
5.2. Experimental Results

Performance Comparison We present federated graph classification results across seven datasets, with the final average test accuracy shown in Tab. 1. The results demonstrate that FedSDR outperforms all baseline methods. Structure-aware approaches like FGSSL and FedGTA exhibit more stable performance across datasets. While MOON achieves competitive results on some datasets by aligning local representations with the global consensus, it suffers significant performance degradation on others due to its inability to repair corrupted graph structure. In addition, methods handling attack and heterogeneity like FedTGE and FedATH fail to identify structural noise. Notably, tradi-



(a) PubMed **(b)** Roman-empire

Figure 4. Test accuracy curves of FedSDR and five other methods on two datasets (PubMed and Roman-empire), with accuracy (%) on the y-axis and communication rounds on the x-axis.



(a) Varying Corruption. **(b)** Parameter α.

Figure 5. Study on varying corruption ratios and noise extents.

tional FL algorithms such as FedAvg and FedProx surpass several complex methods, highlighting how severe structural noise substantially impacts most existing approaches.

Convergence Analysis Fig. 4 illustrates the training curves of the average test accuracy with standard deviation across five random runs of two datasets (PubMed, Roman-empire), comparing FedSDR with various baseline methods. The results demonstrate that existing approaches exhibit significant performance degradation in high-structural-noise scenarios. In contrast, our proposed FedSDR maintains both stable convergence and superior performance, showcasing its robustness to structural noise.

5.3. Varying Corruption Ratios and Noise Extents

We evaluate the robustness of FedSDR under varying degrees of structural corruption on PubMed. As demonstrated in Fig. 5a, FedSDR achieves consistent performance improvements over FedAvg across five representative structural corruption ratios and noise extents, accounting for all potential cases. Our analysis reveals two critical findings: (1) While FedAvg exhibits progressively severe performance degradation with increasing noise extent, FedSDR maintains stable accuracy; (2) Even under high corruption ratios, our adaptive aggregation and local graph reconstruction enable robust collaboration, consistently outperforming the baseline. These results validate the effectiveness of FedSDR in handling diverse structural corruption scenarios.

SNAA	RLSR	Datasets		
		PubMed	Actor	ogbn-products
✗	✗	78.41 ± 0.36	31.28 ± 0.77	72.66 ± 1.53
✓	✗	81.82 ± 0.71	31.56 ± 0.64	74.25 ± 0.32
✗	✓	82.26 ± 0.82	32.07 ± 0.42	75.92 ± 0.59
✓	✓	$\mathbf{82.57 \pm 0.64}$	$\mathbf{32.36 \pm 0.59}$	$\mathbf{78.57 \pm 0.28}$

Table 2. Ablation study of FedSDR.

5.4. Ablation Study

To comprehensively evaluate the contribution of each component in FedSDR, we conducted an ablation study across three datasets. Tab. 2 reveals that SNAA alone improves accuracy over FedAvg through effectively identifying and mitigating structural noise during global aggregation. Furthermore, RLSR achieves noticeable success in reconstructing local corrupted graph structures and restoring message-passing dynamics. The complete FedSDR framework, combining both components, delivers optimal performance by addressing structural noise at both global and local levels. This analysis confirms that both global noise-aware aggregation and local structural reconstruction are indispensable for effective FGL training under structural corruption.

5.5. Hyper-parameter Study

We conduct a thorough investigation of the pruning proportion parameter α to understand its impact on model performance and robustness. As shown in Fig. 5b, FedSDR outperforms FedAvg across all tested α values (ranging from 0.2 to 1.0) on five datasets. Moreover, continuous performance improvement with increasing values of α indicates that our reconstruction successfully distinguishes between corrupted and meaningful edges, enabling effective noise removal while preserving valuable structural relationships. This performance advantage across parameter configurations demonstrates the robustness of our approach.

6. Conclusion

This paper first presents a comprehensive study addressing random structural noise, a disruptive but overlooked challenge in real-world FGL. Existing methods, designed for adversarial attacks or data heterogeneity, fail to address this issue. In response, we propose **FedSDR**, a novel framework that introduces a spectral perspective for integrated noise detection and repair. For global collaboration, **SNAA** evaluates and mitigates structural corruption by dynamically reweighting client contributions based on structural fidelity. For local deployment, **RLSR** reconstructs corrupted graphs by aligning them with the global consensus while preserving valid local patterns. Extensive experiments across multiple datasets and noise scenarios demonstrate the superior robustness of FedSDR, establishing a trustworthy paradigm for FGL in high-structural-noise environments.

7. Acknowledgement

This research is supported by the National Natural Science Foundation of China under Grant (62361166629, 62501428, 625B1007), the Major Project of Science, Technology Innovation of Hubei Province (2024BCA003, 2025BEA002), and Undergraduate Training Programs for Innovation and Entrepreneurship of Wuhan University.

References

[1] Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In *ICML*, pages 1396–1415, 2023. 3, 7

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020. 2

[3] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608. PMLR, 2021. 3

[4] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *International Conference on Learning Representations*, 2021. 3

[5] Dmitry Baranovskiy, Ivan Oseledets, and Andrey Babenko. A critical look at the evaluation of gnns under heterophily: Are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023. 7, 13

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008. 7

[7] Deyu Bo, Yuan Fang, Yang Liu, and Chuan Shi. Graph contrastive learning with stable and scalable spectral encoding. In *NeurIPS*, 2023. 2

[8] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. *arXiv preprint arXiv:2303.01028*, 2023. 3

[9] Khac-Hoai Nam Bui, Jiho Cho, and Hongsuk Yi. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52 (3):2763–2774, 2022. 3

[10] Jinyin Chen, Guohan Huang, Haibin Zheng, Shanqing Yu, Wenrong Jiang, and Chen Cui. Graph-fraudster: Adversarial attacks on graph neural network-based vertical federated learning. *IEEE Transactions on Computational Social Systems*, 10(2):492–506, 2022. 2

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 3

[12] Xiuwen Fang and Mang Ye. Robust federated learning with noisy and heterogeneous clients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10072–10081, 2022. 3, 7

[13] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973. 4

[14] Lele Fu, Bowen Deng, Sheng Huang, Tianchi Liao, Shirui Pan, and Chuan Chen. Less is more: Federated graph learning with alleviating topology heterogeneity from a causal perspective. In *Forty-second International Conference on Machine Learning*. 2, 3, 7

[15] Xingbo Fu, Zihan Chen, Binchi Zhang, Chen Chen, and Jundong Li. Federated graph learning with structure proxy alignment. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 827–838, 2024. 2

[16] Simon Markus Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. Spatio-spectral graph neural networks. *Advances in Neural Information Processing Systems*, 37: 49022–49080, 2024. 3

[17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[18] Mingguo He, Zhewei Wei, Hongteng Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34:14239–14251, 2021. 3

[19] 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. *Advances in neural information processing systems*, 33:22118–22133, 2020. 7, 13

[20] Wenke Huang, Guancheng Wan, Mang Ye, and Bo Du. Federated graph semantic and structural learning. In *IJCAI*, pages 3830–3838, 2023. 7

[21] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 148–156, 2021. 3

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2): 1–210, 2021. 2

[23] Rafeef Kareem and Wesam Bhaya. Fake profiles types of online social networks: a survey. *International Journal of Engineering & Technology*, 7(4.19):919–925, 2018. 2

[24] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 7

[25] D Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *ICLR*, 2015. 7

[26] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph trans-

formers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021. 2

[27] William Leeney and Ryan McConville. A framework for exploring federated community detection. *arXiv preprint arXiv:2312.09023*, 2023. 3

[28] Ang Li, Yawen Li, Zhe Xue, Zeli Guan, and Mengyu Zhuang. Learning hierarchy-aware federated graph embedding for link prediction. In *2024 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 329–336. IEEE, 2024. 3

[29] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020. 1

[30] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 7

[31] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2020. 7

[32] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368, 2021. 7

[33] Xunkai Li, Zhengyu Wu, Wentao Zhang, Henan Sun, Rong-Hua Li, and Guoren Wang. Adafgl: A new paradigm for federated node classification with topology heterogeneity. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 2517–2530. IEEE, 2024. 2, 3, 7

[34] Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. In *VLDB*, 2024. 3, 7

[35] Yang Li, Michael Purcell, Thierry Rakotoarivelo, David Smith, Thilina Ranbaduge, and Kee Siong Ng. Private graph data release: A survey. *ACM Computing Surveys*, 55(11): 1–39, 2023. 1

[36] Rui Liu and Han Yu. Federated graph neural networks: Overview, techniques and challenges. *arXiv preprint arXiv:2202.07256*, 2022. 3

[37] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022. 1

[38] Weimin Lyu, Songzhu Zheng, Lu Pang, Haibin Ling, and Chao Chen. Attention-enhancing backdoor attacks against bert-based models. *arXiv preprint arXiv:2310.14480*, 2023. 2

[39] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 7

[40] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations, ICLR*, 2020. 7, 13

[41] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008. 1

[42] Marco Schreyer, Timur Sattarov, and Damian Borth. Federated and privacy-preserving learning of accounting data in financial statement audits. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 105–113, 2022. 1

[43] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 7, 12

[44] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018. 1, 7, 12

[45] Kimberly Stachenfeld, Jonathan Godwin, and Peter Battaglia. Graph networks with spectral message passing. *arXiv preprint arXiv:2101.00079*, 2020. 3

[46] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8432–8440, 2022. 7

[47] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *AAAI*, pages 9953–9961, 2023. 3

[48] Zihan Tan, Guancheng Wan, Wenke Huang, and Mang Ye. FedSSP: Federated graph learning with spectral knowledge and personalized preference. *arXiv preprint arXiv:2410.20105*, 2024. 1, 2, 7

[49] Zihan Tan, Suyuan Huang, Guancheng Wan, Wenke Huang, He Li, and Mang Ye. S2FGL: Spatial spectral federated graph learning. *arXiv preprint arXiv:2507.02409*, 2025. 3

[50] Zihan Tan, Guancheng Wan, Wenke Huang, He Li, Guibin Zhang, Carl Yang, and Mang Ye. FedSPA: Generalizable federated graph learning under homophily heterogeneity. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15464–15475, 2025. 1, 3

[51] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 3

[52] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 4

[53] Guancheng Wan, Zitong Shi, Wenke Huang, Guibin Zhang, Dacheng Tao, and Mang Ye. Energy-based backdoor defense against federated graph learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 7

[54] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International conference on machine learning*, pages 23341–23362. PMLR, 2022. 3

[55] Chuhan Wu, Fangzhao Wu, Yang Cao, Yongfeng Huang, and Xing Xie. Fedgnn: Federated graph neural network

for privacy-preserving recommendation. *arXiv preprint arXiv:2102.04925*, 2021. 1

[56] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):3091, 2022. 1

[57] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020. 3

[58] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. 3

[59] Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33:17009–17021, 2020. 3

[60] Wentao Yu, Shuo Chen, Yongxin Tong, Tianlong Gu, and Chen Gong. Modeling inter-intra heterogeneity for graph federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22236–22244, 2025. 3, 7

[61] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–803, 2019. 3

[62] Ke Zhang, Carl Yang, Xiaoxiao Li, Lichao Sun, and Siu Ming Yiu. Subgraph federated learning with missing neighbor generation. *Advances in neural information processing systems*, 34:6671–6682, 2021. 2, 7

[63] Yi Zhang, Yuying Zhao, Zhaoqing Li, Xueqi Cheng, Yu Wang, Olivera Kotevska, S Yu Philip, and Tyler Derr. A survey on privacy in graph neural networks: Attacks, preservation, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2024. 1

[64] Dingyi Zhao. FedSST: Rethinking fair federated graph learning under structural shift. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2026. 1

[65] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020. 3

[66] Banghua Zhu, Lun Wang, Qi Pang, Shuai Wang, Jiantao Jiao, Dawn Song, and Michael I Jordan. Byzantine-robust federated learning with optimal statistical rates. In *International Conference on Artificial Intelligence and Statistics*, pages 3151–3178. PMLR, 2023. 3

[67] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11168–11176, 2021. 1

[68] Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: topology-aware data-free knowledge distillation for subgraph federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 5716–5724, 2024. 7

A. Computational Overhead Analysis

While our method achieves strong performance, we acknowledge that the spectral analysis in SNAA and graph repair in RLSR introduce modest computational overhead. Nevertheless, the overhead can be greatly reduced under GPU batch processing, sparse matrix multiplication and multiple clients partition, which makes our method practical on large graphs. For a k-layer GNN with batch size b and feature dimension f, T denotes moments order, while linear parameters include N, M, c (nodes, edges, classes), s (features aggregation steps), g (generated neighbors), K (selected clients per round), d (selected augmented nodes). We note that in our complexity analysis, N and M refer to the entire graph for server-side operations, but are scaled to the local subgraph size for client-side computations. For scenarios with extremely large local subgraphs, one could adopt approximate k' nearest-neighbor search to reduce client memory and time complexity to $\mathbf{O(Nk')}$ and $\mathbf{O(Nk'f)}$, where $k' << N$. Thus, the computation of the local similarity matrix is highly efficient in practice. Analysis in Tab. 4 shows that FedSDR achieves top performance with reasonable overhead.

In addition, we analyze the computational complexity of the key operation $S_{\text{ide}} = \frac{\mathbf{D}^T\mathbf{LD}}{\mathbf{D}^T\mathbf{D}}$ by leveraging the Compressed Sparse Row (CSR) format for sparse matrix multiplication. In the CSR format, the dominant computational cost of multiplying two sparse matrices $\mathbf{E}$ and $\mathbf{F}$ arises from the total number of inner-loop iterations, denoted as I. Let $n_{\mathbf{E}}$ and $n_{\mathbf{F}}$ denote the total number of non-zero entries in $\mathbf{E}$ and $\mathbf{F}$, respectively. This quantity I can be expressed as:

$$I = \sum_{i=0}^{N-1} r_{\mathbf{F}}(i) \cdot c_{\mathbf{E}}(i),$$

where $r_{\mathbf{F}}(i)$ denotes the number of non-zero entries in row i of $\mathbf{F}$, and $c_{\mathbf{E}}(i)$ denotes the number of non-zero entries in column i of $\mathbf{E}$. Noting that $\sum_{i=0}^{N-1} c_{\mathbf{E}}(i) = n_{\mathbf{E}}$, and assuming a relatively uniform distribution of non-zero entries across the rows of $\mathbf{F}$ such that $r_{\mathbf{F}}(i) \approx n_{\mathbf{F}}/N$, we obtain the approximation:

$$I \approx n_{\mathbf{E}} \cdot \frac{n_{\mathbf{F}}}{N},$$

which yields a computational complexity of $O(n_{\mathbf{E}} \cdot n_{\mathbf{F}}/N)$.

Complexity of $\mathbf{D}^T\mathbf{LD}$. We analyze the computation in two sparse matrix multiplications. The degree matrix $\mathbf{D}$ is diagonal, so $n_{\mathbf{D}} = n_{\mathbf{D}^T} = N$. For the adjacency matrix $\mathbf{A}$ of an undirected graph without self-loops, the handshaking lemma gives $n_{\mathbf{A}} = 2M$, and therefore the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ contains $n_{\mathbf{L}} = n_{\mathbf{A}} + n_{\mathbf{D}} = 2M + N$ non-zero entries. Since directed graphs or the presence of self-loops would reduce the values of $n_{\mathbf{A}}$ and $n_{\mathbf{L}}$, the undirected graph without self-loops represents the worst-case computation.

- **Step 1: Compute $\mathbf{G} = \mathbf{D}^T\mathbf{L}$** This is a left multiplication by a diagonal matrix. The complexity is:

$$O\left(n_{\mathbf{D}^T} \cdot \frac{n_{\mathbf{L}}}{N}\right) = O\left(N \cdot \frac{2M+N}{N}\right) = O(2M+N).$$

  The resulting matrix $\mathbf{G}$ inherits the sparsity pattern of $\mathbf{L}$, so $n_{\mathbf{G}} = 2M + N$.
- **Step 2: Multiply $\mathbf{G}$ with $\mathbf{D}$.** This is a right multiplication by a diagonal matrix. The complexity is:

$$O\left(n_{\mathbf{G}} \cdot \frac{n_{\mathbf{D}}}{N}\right) = O\left((2M+N) \cdot \frac{N}{N}\right) = O(2M+N).$$

Thus, the total complexity for computing $\mathbf{D}^T\mathbf{LD}$ is $O(2M+N) + O(2M+N) = O(M+N)$.

Complexity of $\mathbf{D}^T\mathbf{D}$. Since both matrices are diagonal, their multiplication takes:

$$O\left(n_{\mathbf{D}^T} \cdot \frac{n_{\mathbf{D}}}{N}\right) = O\left(N \cdot \frac{N}{N}\right) = O(N).$$

Complexity of Summation and Division. Summing all non-zero entries of $\mathbf{D}^T\mathbf{LD}$ requires $O(M+N)$ time, and summing the N diagonal entries of $\mathbf{D}^T\mathbf{D}$ requires $O(N)$. The final scalar division costs $O(1)$. Therefore, the total complexity is $O(M+N) + O(N) + O(1) = O(M+N)$.

Overall Complexity. Combining all components yields:

$$O(M+N) + O(N) + O(M+N) = O(M+N).$$

This linear time complexity of S_{ide} with respect to the graph size confirms the practical efficiency of our method for large-scale graphs.

B. Datasets

Statistics and configurations of the datasets used in our experiments are provided in Tab. 5.

PubMed: PubMed [43] is a widely adopted citation network dataset in which nodes represent academic papers and edges correspond to citation relationships between them. Each node is associated with a word vector feature encoding the presence or absence of specific keywords in the corresponding paper. As a standard benchmark in graph-based machine learning, this dataset is commonly employed for node classification tasks, particularly in FL scenarios where data privacy must be preserved.

Coauthor-CS, Coauthor-Physics: Derived from the Microsoft Academic Graph, the Coauthor-CS and Coauthor-Physics datasets [44] originate from the KDD Cup 2016 challenge. These benchmark datasets represent academic collaboration networks, with nodes corresponding to authors and edges indicating co-authorship. Coauthor-CS comprises 18,333 nodes and 81,894 edges, featuring node attributes based on paper keywords and 15 distinct research

fields as class labels. The larger Coauthor-Physics dataset contains 34,493 nodes and 247,962 edges, with similar node features and labels representing classification into 5 main research areas. Both datasets serve as established benchmarks for evaluating graph neural networks, particularly for node classification tasks, due to their realistic academic network structures and comprehensive feature representations.

Actor: The Actor dataset [40] is an actor co-occurrence network from Wikipedia, where nodes represent actors and edges indicate co-appearance in Wikipedia pages. Each node is characterized by a bag-of-words feature vector derived from the corresponding Wikipedia page content. The dataset includes five actor categories, determined through semantic analysis of their associated Wikipedia entries. As a standard benchmark in graph machine learning, it is widely used to evaluate node classification tasks and related graph-based learning problems.

Roman-empire: The Roman-empire dataset [5] is derived from the English Wikipedia article about the Roman Empire. Nodes represent word occurrences (including non-unique words), with the graph size reflecting the article length. Edges are based on two linguistic relationships: (1) sequential co-occurrence when words appear consecutively in the text, and (2) syntactic dependencies from the sentence parse trees. This construction yields a chain-like graph augmented with linguistic connections, capturing sequential and syntactic relationships between words.

ogbn-mag: The ogbn-mag dataset from the Open Graph Benchmark (OGB) facilitates node property prediction within a heterogeneous academic network derived from the Microsoft Academic Graph (MAG). This graph comprises four entity types—papers (736,389 nodes), authors (1.13 million nodes), institutions (8,740 nodes), and fields of study (59,965 nodes)—interconnected by directed relations such as authorship, citation, and affiliation. Each paper node possesses a 128-dimensional word2vec feature vector, while other lack initial features. The objective is a 349-class classification to predict the publishing venue of each paper. A temporally realistic split is employed, where papers published before 2018 are used for training, those from 2018 for validation, and papers since 2019 for testing, forecasting future trends based on historical data [19].

ogbn-products: The ogbn-products dataset from the Open Graph Benchmark (OGB) is designed for node property prediction within an Amazon product co-purchasing network. This large-scale, undirected, and unweighted graph contains approximately 2.4 million product nodes and 61.9 million edges representing co-purchase relationships. Each node is associated with a 100-dimensional feature vector generated from product descriptions using bag-of-words and PCA. The objective is a multi-class classification task to predict one of 47 top-level product categories. Reflecting a realistic scenario, the dataset is split by sales rank: the top 8% of popular products are used for training, the subsequent 2% for validation, and the remaining 90% for testing, thereby requiring predictions for less popular items [19].

C. Privacy Analysis

To analyze the privacy-utility trade-off, we evaluate FedSDR on PubMed under varying noise multipliers σ, keeping the gradient clipping threshold B and the failure probability δ fixed at 1.0 and 10^{-5}, respectively. As shown in Tab. 3, FedSDR maintains robust performance even under strict privacy guarantees. Interestingly, we observe a **slight performance improvement**, which suggests that the gradient clipping, combined with our structure repair, effectively prevents overfitting to structural noise.

Metric	No DP	Weak DP ($\sigma = 0.5$)	Strong DP ($\sigma = 1.0$)
Acc	82.57	**83.69** $\uparrow$1.12	**83.46** $\uparrow$0.89

Table 3. Accuracy under differential privacy (DP) constraints.

D. Additional Experiments

We comprehensively compare FedSDR with state-of-the-art methods under varying **corruption ratios** (the proportion of structurally corrupted clients) and **noise extents** (the proportion of edges randomly added and then deleted from the original graph in corrupted clients). Our experimental setup evaluates three corruption ratios of 0.3, 0.5, and 1 (the extreme case detailed in Tab. 1), with the noise extent and the pruning proportion α fixed at 0.5 and 0.3. In addition, we validate the robustness of FedSDR across three noise extents (0.3, 0.5, and 1), while keeping the corruption ratio at 0.5 and the pruning proportion α at 0.3 constant. Notably, as the corruption severity increases, competing methods exhibit significant performance degradation, whereas FedSDR maintains stable performance, demonstrating a substantial advantage. The best and second results are highlighted with bold and underline, respectively.

Methods	Client Mem	Server Mem	Client Time	Server Time
FedGTA	$\mathbf{O((b+s)f+f^2+sTc)}$	$O(Kf^2+KsTc)$	$O(sM(f+sNc)+N(f^2+c))$	$O(Kf+KsTc)$
FedTAD	$O((b+s)f+f^2+Nf+kMf)$	$O(Kf^2+dgf+kf^2)$	$\mathbf{O(sNMf+Nf^2)}$	$O(Kf+(k+N)f^2+kdgf)$
FedSDR	$O(N^2+f^2+kMf)$	$\mathbf{O(K+Kf^2)}$	$O(M+N+kMf+Nf^2+N^2f)$	$\mathbf{O(Kf+K)}$

Table 4. Complexity analysis among structure-related methods. Best in bold.

Dataset	Nodes	Edges	Classes	Features	Learning Rate	Rounds	Clients
PubMed	19,717	44,338	3	500	0.01	1,000	10
Coauthor-CS	18,333	81,894	15	6,805	0.005	1,000	50
Coauthor-Phy	34,493	247,962	5	8,415	0.01	1,000	100
Actor	7,600	29,926	5	931	0.002	1,000	20
Roman-empire	22,662	32,927	18	300	0.02	5,000	50
ogbn-mag	1,939,743	21,111,007	349	128	0.01	1,000	100
ogbn-products	2,449,029	61,859,140	47	100	0.01	1,000	100

Table 5. Statistics and configurations of datasets used in experiments.

Methods	PubMed	Coauthor-CS	Coauthor-Phy	Actor	Roman-empire	ogbn-mag	ogbn-products
FedAvg [ASTAT17]	81.83 ± 0.69	81.69 ± 0.47	89.08 ± 0.27	31.37 ± 0.58	39.23 ± 0.50	40.79 ± 0.57	73.34 ± 0.49
FedProx [arxiv18]	82.23 ± 0.68	82.05 ± 0.22	89.50 ± 0.29	$\underline{31.57 \pm 0.65}$	39.74 ± 0.54	$\underline{42.47 \pm 0.46}$	73.06 ± 0.47
Scaffold [ICML20]	40.45 ± 1.81	48.57 ± 2.43	79.88 ± 0.44	23.69 ± 0.46	17.90 ± 0.32	14.52 ± 0.42	66.28 ± 0.41
MOON [CVPR21]	76.65 ± 2.76	$\underline{83.36 \pm 0.18}$	$\mathbf{91.06 \pm 0.16}$	31.36 ± 0.69	$\underline{39.97 \pm 0.17}$	40.33 ± 0.37	73.47 ± 0.36
Ditto [ICML21]	78.92 ± 0.89	81.28 ± 0.75	88.61 ± 0.49	30.71 ± 0.27	38.26 ± 0.33	41.08 ± 0.79	71.95 ± 0.82
FedSage+ [NIPS21]	81.73 ± 0.68	81.92 ± 0.23	89.11 ± 1.02	30.95 ± 0.52	39.61 ± 0.69	39.89 ± 0.41	72.85 ± 0.40
FedProto [AAAI22]	82.01 ± 0.40	74.14 ± 0.29	86.70 ± 0.23	23.45 ± 0.29	13.93 ± 0.48	35.21 ± 0.56	69.20 ± 0.39
RHFL [CVPR22]	82.03 ± 0.53	82.18 ± 0.66	89.08 ± 0.41	31.06 ± 0.26	39.88 ± 0.45	42.04 ± 0.62	$\underline{74.58 \pm 0.57}$
FGSSL [IJCAI23]	82.09 ± 0.41	80.97 ± 0.57	89.94 ± 0.33	30.97 ± 0.52	33.16 ± 0.32	41.26 ± 0.63	72.32 ± 0.24
FED-PUB [ICML23]	80.27 ± 0.54	80.84 ± 0.56	87.90 ± 0.34	28.96 ± 0.73	38.39 ± 0.47	37.47 ± 0.29	71.53 ± 0.27
FedTAD [IJCAI24]	$\underline{82.62 \pm 0.43}$	73.16 ± 1.55	OOM	30.79 ± 0.45	39.83 ± 0.80	OOM	OOM
AdaFGL [ICDE24]	81.32 ± 0.71	82.17 ± 0.37	88.75 ± 0.48	31.21 ± 0.59	39.28 ± 0.50	40.36 ± 0.78	72.94 ± 0.66
FedGTA [VLDB24]	82.20 ± 0.75	81.03 ± 0.60	85.77 ± 0.16	31.30 ± 0.50	38.80 ± 1.12	38.74 ± 0.69	73.78 ± 0.53
FedSSP [NIPS24]	79.90 ± 0.32	81.22 ± 0.39	88.04 ± 0.50	30.74 ± 0.81	37.52 ± 0.65	37.25 ± 0.47	72.02 ± 0.35
FedTGE [ICLR25]	78.89 ± 0.73	78.17 ± 0.80	87.25 ± 0.92	29.48 ± 0.79	36.26 ± 0.37	38.35 ± 0.86	70.23 ± 0.83
FedATH [ICML25]	78.04 ± 0.55	79.84 ± 0.49	87.63 ± 0.61	30.29 ± 0.46	36.43 ± 0.41	38.19 ± 0.39	71.88 ± 0.76
FedIIH [AAAI25]	79.32 ± 0.46	80.87 ± 0.42	88.16 ± 0.59	30.43 ± 0.67	38.65 ± 0.53	38.94 ± 0.40	73.36 ± 0.54
FedSDR (ours)	$\mathbf{84.07 \pm 0.48}$	$\mathbf{83.43 \pm 0.39}$	$\underline{90.72 \pm 0.43}$	$\mathbf{33.30 \pm 0.42}$	$\mathbf{41.58 \pm 0.38}$	$\mathbf{46.88 \pm 0.21}$	$\mathbf{80.57 \pm 0.42}$

Table 6. Corruption Ratio $= 0.3$, Noise Extent $= 0.5$, $\alpha = 0.3$

Methods	PubMed	Coauthor-CS	Coauthor-Phy	Actor	Roman-empire	ogbn-mag	ogbn-products
FedAvg [ASTAT17]	81.28 ± 2.02	81.26 ± 0.26	88.20 ± 0.22	31.29 ± 0.53	39.14 ± 0.60	41.58 ± 0.81	72.85 ± 0.24
FedProx [arxiv18]	80.82 ± 0.96	82.48 ± 0.51	88.68 ± 0.13	31.56 ± 0.59	39.26 ± 0.50	42.06 ± 0.27	72.83 ± 0.87
Scaffold [ICML20]	40.71 ± 0.90	44.72 ± 4.54	80.21 ± 1.61	23.06 ± 1.37	16.76 ± 1.86	14.07 ± 2.02	62.51 ± 2.03
MOON [CVPR21]	77.09 ± 2.59	80.92 ± 0.54	90.08 ± 0.17	31.04 ± 0.64	40.41 ± 0.22	42.34 ± 0.52	72.07 ± 0.22
Ditto [ICML21]	77.95 ± 0.85	80.31 ± 0.82	86.91 ± 0.32	30.03 ± 0.28	38.32 ± 0.24	42.85 ± 1.01	71.73 ± 0.25
FedSage+ [NIPS21]	79.91 ± 0.82	81.38 ± 0.57	87.30 ± 0.81	31.42 ± 0.43	39.03 ± 0.57	41.11 ± 0.47	72.36 ± 0.38
FedProto [AAAI22]	79.95 ± 0.58	72.46 ± 0.43	85.55 ± 0.21	23.50 ± 0.28	15.94 ± 0.13	36.04 ± 0.59	68.13 ± 0.44
RHFL [CVPR22]	79.86 ± 0.67	82.07 ± 0.49	87.34 ± 0.35	31.40 ± 0.48	38.33 ± 0.45	42.31 ± 0.92	72.76 ± 0.29
FGSSL [IJCAI23]	78.36 ± 0.46	81.48 ± 0.42	88.60 ± 0.76	31.31 ± 0.30	34.58 ± 0.56	40.38 ± 0.30	71.64 ± 0.34
FED-PUB [ICML23]	76.82 ± 0.59	79.79 ± 0.37	86.28 ± 0.31	30.06 ± 0.35	37.12 ± 0.65	39.51 ± 0.56	71.22 ± 0.46
FedTAD [IJCAI24]	80.93 ± 0.58	74.81 ± 1.38	OOM	31.25 ± 0.57	40.53 ± 0.46	OOM	OOM
AdaFGL [ICDE24]	78.74 ± 0.60	80.06 ± 0.74	87.90 ± 0.65	31.06 ± 0.32	40.36 ± 0.63	41.67 ± 0.30	71.24 ± 0.65
FedGTA [VLDB24]	81.30 ± 0.77	81.22 ± 0.49	88.28 ± 0.18	31.25 ± 0.22	39.64 ± 0.76	41.12 ± 0.35	72.01 ± 0.63
FedSSP [NIPS24]	78.24 ± 0.81	80.12 ± 0.71	86.56 ± 0.92	30.58 ± 0.73	38.15 ± 0.62	40.21 ± 0.58	71.81 ± 0.32
FedTGE [ICLR25]	78.22 ± 1.18	77.62 ± 0.84	86.13 ± 0.91	31.05 ± 0.88	37.04 ± 0.79	38.31 ± 0.37	69.57 ± 0.90
FedATH [ICML25]	77.42 ± 0.63	79.08 ± 0.73	86.65 ± 0.50	30.11 ± 0.53	37.70 ± 0.50	39.87 ± 0.46	69.85 ± 0.48
FedIIH [AAAI25]	78.67 ± 0.96	80.15 ± 0.75	87.34 ± 0.65	29.94 ± 0.76	38.86 ± 0.52	41.35 ± 0.68	72.83 ± 0.51
FedSDR (ours)	$\mathbf{84.06 \pm 0.44}$	$\mathbf{83.10 \pm 0.32}$	$\mathbf{90.50 \pm 0.23}$	$\mathbf{33.10 \pm 0.50}$	$\mathbf{45.12 \pm 0.36}$	$\mathbf{48.06 \pm 0.20}$	$\mathbf{80.23 \pm 1.20}$

Table 7. Corruption Ratio $= 0.5$, Noise Extent $= 0.5$, $\alpha = 0.3$

Methods	PubMed	Coauthor-CS	Coauthor-Phy	Actor	Roman-empire	ogbn-mag	ogbn-products
FedAvg [ASTAT17]	82.60 ± 0.22	81.47 ± 0.27	88.97 ± 0.22	31.07 ± 0.47	38.82 ± 0.42	41.07 ± 0.61	72.68 ± 0.35
FedProx [arxiv18]	82.44 ± 0.65	81.87 ± 0.26	89.60 ± 0.52	31.47 ± 0.43	39.49 ± 0.84	41.09 ± 0.38	72.50 ± 0.40
Scaffold [ICML20]	41.90 ± 0.41	50.70 ± 1.58	79.61 ± 2.18	22.07 ± 0.98	17.57 ± 0.33	13.77 ± 1.22	64.76 ± 1.05
MOON [CVPR21]	76.68 ± 3.39	82.09 ± 0.45	90.40 ± 0.37	31.85 ± 1.31	40.17 ± 0.26	42.56 ± 0.42	72.01 ± 0.31
Ditto [ICML21]	78.32 ± 0.82	81.24 ± 0.45	87.39 ± 0.43	30.50 ± 0.30	38.46 ± 0.68	42.34 ± 0.80	71.92 ± 0.55
FedSage+ [NIPS21]	80.50 ± 0.70	81.60 ± 0.52	88.56 ± 0.83	31.00 ± 0.40	40.10 ± 0.65	40.57 ± 0.40	72.23 ± 0.46
FedProto [AAAI22]	81.30 ± 0.43	74.27 ± 0.31	86.42 ± 0.14	23.04 ± 1.07	14.52 ± 0.19	35.66 ± 0.58	68.60 ± 0.41
RHFL [CVPR22]	80.06 ± 0.60	81.51 ± 0.74	88.23 ± 0.92	31.22 ± 0.45	38.75 ± 0.53	42.35 ± 0.78	73.24 ± 0.59
FGSSL [IJCAI23]	81.39 ± 0.51	80.51 ± 0.30	89.95 ± 0.24	30.88 ± 0.68	33.95 ± 0.80	41.15 ± 0.36	72.25 ± 0.32
FED-PUB [ICML23]	78.12 ± 0.68	80.34 ± 0.42	86.89 ± 0.43	29.28 ± 0.33	38.50 ± 0.65	39.22 ± 0.55	71.47 ± 0.50
FedTAD [IJCAI24]	82.67 ± 0.31	73.25 ± 1.66	OOM	30.93 ± 0.98	40.30 ± 0.37	OOM	OOM
AdaFGL [ICDE24]	79.86 ± 0.72	81.50 ± 0.61	88.45 ± 0.58	31.15 ± 0.42	39.34 ± 0.67	40.21 ± 0.43	72.63 ± 0.64
FedGTA [VLDB24]	82.19 ± 0.45	81.26 ± 0.27	88.87 ± 0.07	31.55 ± 0.30	40.22 ± 0.40	40.80 ± 0.60	72.93 ± 0.40
FedSSP [NIPS24]	79.17 ± 0.70	80.50 ± 0.60	87.79 ± 0.70	30.70 ± 0.70	37.26 ± 0.60	38.58 ± 0.50	71.83 ± 0.51
FedTGE [ICLR25]	78.48 ± 0.67	77.95 ± 0.52	86.39 ± 0.74	30.16 ± 0.41	36.38 ± 0.63	37.29 ± 0.58	69.90 ± 0.49
FedATH [ICML25]	77.73 ± 0.48	79.21 ± 0.61	87.14 ± 0.53	30.43 ± 0.37	36.68 ± 0.44	38.02 ± 0.39	70.87 ± 0.62
FedIIH [AAAI25]	78.81 ± 0.55	80.60 ± 0.47	87.75 ± 0.68	31.19 ± 0.52	37.95 ± 0.51	39.26 ± 0.43	72.40 ± 0.57
FedSDR (ours)	$\mathbf{84.57 \pm 0.49}$	$\mathbf{82.82 \pm 0.32}$	$\mathbf{90.45 \pm 0.14}$	$\mathbf{33.37 \pm 0.50}$	$\mathbf{44.36 \pm 0.21}$	$\mathbf{47.50 \pm 0.30}$	$\mathbf{80.80 \pm 0.40}$

Table 8. Corruption Ratio $= 0.5$, Noise Extent $= 0.3$, $\alpha = 0.3$

Methods	PubMed	Coauthor-CS	Coauthor-Phy	Actor	Roman-empire	ogbn-mag	ogbn-products
FedAvg [ASTAT17]	78.89 ± 0.21	76.27 ± 0.76	86.83 ± 0.50	30.97 ± 0.50	39.15 ± 0.71	42.26 ± 0.65	$\underline{72.43 \pm 0.82}$
FedProx [arxiv18]	78.59 ± 1.38	76.35 ± 0.07	87.05 ± 0.68	31.54 ± 0.53	38.77 ± 0.37	41.28 ± 0.37	72.14 ± 0.64
Scaffold [ICML20]	42.60 ± 1.50	42.46 ± 0.90	77.69 ± 1.16	24.02 ± 0.34	17.91 ± 0.36	17.29 ± 2.15	59.43 ± 1.97
MOON [CVPR21]	71.84 ± 1.66	$\underline{76.91 \pm 0.68}$	$\underline{88.59 \pm 0.55}$	31.55 ± 0.83	39.66 ± 0.22	41.83 ± 0.72	69.86 ± 0.51
Ditto [ICML21]	77.37 ± 0.91	75.76 ± 0.67	86.97 ± 0.43	30.95 ± 0.31	39.71 ± 0.52	41.15 ± 0.87	71.44 ± 0.73
FedSage+ [NIPS21]	78.35 ± 0.78	75.04 ± 0.56	87.28 ± 0.64	30.68 ± 0.49	40.58 ± 0.61	41.52 ± 0.46	70.55 ± 0.39
FedProto [AAAI22]	78.88 ± 1.06	69.41 ± 0.24	83.72 ± 0.20	23.38 ± 0.67	17.55 ± 0.25	36.45 ± 0.63	68.42 ± 0.87
RHFL [CVPR22]	78.43 ± 0.72	75.87 ± 0.81	86.69 ± 0.57	31.33 ± 0.54	$\underline{41.36 \pm 0.48}$	$\underline{42.54 \pm 0.63}$	72.11 ± 0.45
FGSSL [IJCAI23]	$\underline{79.04 \pm 1.95}$	74.50 ± 0.75	88.10 ± 0.61	31.63 ± 1.03	33.29 ± 0.20	42.27 ± 0.54	70.81 ± 0.76
FED-PUB [ICML23]	75.71 ± 0.62	76.56 ± 0.41	85.41 ± 0.58	30.72 ± 0.37	39.15 ± 0.59	39.76 ± 0.52	70.66 ± 0.83
FedTAD [IJCAI24]	78.68 ± 0.87	70.56 ± 1.41	OOM	31.23 ± 0.47	39.56 ± 0.41	OOM	OOM
AdaFGL [ICDE24]	77.63 ± 0.88	76.77 ± 0.65	87.53 ± 0.49	31.48 ± 0.42	41.06 ± 0.57	42.22 ± 0.34	70.43 ± 0.71
FedGTA [VLDB24]	78.67 ± 3.10	73.59 ± 0.37	86.60 ± 0.50	$\underline{31.90 \pm 0.55}$	38.72 ± 0.35	41.85 ± 0.41	71.76 ± 0.68
FedSSP [NIPS24]	77.83 ± 0.54	75.26 ± 0.47	86.21 ± 0.68	31.24 ± 0.61	40.42 ± 0.53	40.71 ± 0.44	69.17 ± 0.79
FedTGE [ICLR25]	76.50 ± 0.73	73.63 ± 0.59	85.75 ± 0.81	30.49 ± 0.46	38.44 ± 0.67	39.80 ± 0.54	68.84 ± 0.71
FedATH [ICML25]	77.17 ± 0.42	76.42 ± 0.57	86.52 ± 0.49	30.28 ± 0.38	38.77 ± 0.41	40.43 ± 0.47	69.50 ± 0.53
FedIIH [AAAI25]	78.51 ± 0.64	76.25 ± 0.50	87.19 ± 0.72	31.04 ± 0.43	37.86 ± 0.56	41.71 ± 0.52	71.95 ± 0.60
FedSDR (ours)	$\mathbf{83.75 \pm 0.25}$	$\mathbf{80.64 \pm 0.39}$	$\mathbf{89.52 \pm 0.24}$	$\mathbf{33.50 \pm 0.89}$	$\mathbf{46.40 \pm 0.61}$	$\mathbf{46.73 \pm 0.33}$	$\mathbf{79.36 \pm 0.47}$

Table 9. Corruption Ratio $= 0.5$, Noise Extent $= 1$, $\alpha = 0.3$