International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 10 Issue 2, Mar-Apr 2026 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Adaptive Normalization and Aggregation Perturbation for GNNs on Non-Uniform Data Distributions

Tianjiao Luo, Xinru Gao, Yuhang Liu, Jiayu Hou, Zhouwan, Guoqing, Haoran Liu, Dingdang, Weixin, Yucheng Tian, Xiangchen Li

School of Systems Science and Statistics, Beijing Wuzi University, Beijing, China

ABSTRACT

Graph neural networks (GNNs) tend to suffer from lower robustness and reduced prediction accuracy when trained under differential privacy (DP), especially when the graph data follow a non-uniform distribution or contain outliers. Traditional aggregation-based perturbation methods often fail to adapt to such complex data characteristics, and they struggle to find a proper balance between privacy protection and model performance. To tackle these problems, this paper introduces an adaptive normalization and aggregation perturbation approach named MVNAP, specifically designed for non-uniform data distributions. The proposed method consists of three main components: feature-wise personalized normalization using mean and variance, aggregation over sparse graph node matrices, and Gaussian differential privacy perturbation. This design removes scale deviations across different feature dimensions at the root level, ensures strict privacy protection, and retains the essential structural information of the graph. MVNAP has a decoupled, modular architecture, allowing it to be seamlessly integrated as a plug-and-play module into various DP-GNN frameworks without modifying the original network structure or training logic. We evaluate the method on benchmark datasets with power-law distributions and varying feature scales—Reddit, Amazon, and FB-100—under both edge-level and node-level DP scenarios. Experimental results show that the proposed DP-GNN method is well suited for complex, non-uniform data distributions and can serve as a key optimization module to facilitate the practical deployment of differentially private graph neural networks in privacy-sensitive applications.

KEYWORDS: *Graph Neural Networks, Differential Privacy, Aggregation Perturbation, Non-Uniform Data Distribution, Model Robustness.*

How to cite this paper: Tianjiao Luo | Xinru Gao | Yuhang Liu | Jiayu Hou | Zhouwan | Guoqing | Haoran Liu | Dingdang | Weixin | Yucheng Tian | Xiangchen Li "Adaptive Normalization and Aggregation Perturbation for GNNs on Non-Uniform Data Distributions" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-10 | Issue-2, April 2026, pp.896-904, URL: www.ijtsrd.com/papers/ijtsrd116444.pdf

IJTSRD116444

1. INTRODUCTION

Graph Neural Networks (GNNs) leverage their neighborhood information aggregation and message-passing mechanisms to capture both topological and node-feature correlations in graph-structured data. As a result, they have become a core technology for processing complex relational data across domains such as social network analysis [1], recommendation systems [2], and biomolecular network modeling [3]. However, the deep modeling of user node connection patterns and attribute features inherently carries the risk of leaking sensitive information. For example, attackers can extract private individual data from a trained model through membership inference [4], attribute inference, or model inversion [4–6], which severely limits the deployment of GNNs in privacy-sensitive scenarios.

Differential Privacy (DP) provides a rigorous theoretical foundation and a practical implementation path for protecting GNN privacy. It injects controllable random noise during data processing and model training, ensuring that the presence or absence of any single data point does not significantly affect the output [7]. Recent progress in DP-GNN research has

produced methods for edge-level, node-level, and dual-layer privacy protection. Among these, aggregation perturbation—a core DP-GNN technique—perturbs the aggregated features of node neighborhoods to protect node and edge privacy while preserving key graph structural information, making it a current research focus [8]. Notable examples include the GAP framework, which first achieved dual edge- and node-level privacy guarantees, and PROGAP, which optimizes the privacy-performance trade-off via a progressive learning strategy. However, the traditional Normalized Aggregation Perturbation (NAP) method used by both frameworks has fundamental limitations.

Real-world graph data often exhibit non-uniform distributions, such as power-law node degree distributions in social networks, scale discrepancies across features, and outliers introduced during data collection. Traditional aggregation perturbation methods struggle to adapt to these complexities: (1) NAP relies on global uniform normalization without personalized adjustments to the mean and variance of each feature dimension, which fails to fully remove scale biases and leads to imbalanced sensitivity in aggregated features; (2) its "aggregate-then-perturb" pipeline treats normalization as a secondary step, making it difficult to support uniform privacy noise injection and worsening the privacy-performance trade-off; (3) the lack of robustness against outliers allows outliers to propagate through neighborhood aggregation and interact with privacy noise, further degrading model robustness and prediction accuracy.

To overcome these challenges, there is a clear need for DP-GNN aggregation perturbation methods tailored to non-uniform data distributions, with an emphasis on balancing data adaptability and the privacy-performance trade-off. This paper proposes a Mean-Variance Normalized Aggregation Perturbation (MVNAP) method, which integrates personalized normalization, graph node aggregation, and Gaussian privacy perturbation to adaptively process non-uniformly distributed graph data under differential privacy. MVNAP has three core advantages: (1) personalized per-feature mean-variance normalization eliminates feature scale biases at the source and reduces the impact of outliers; (2) the seamless "normalization-aggregation-perturbation" pipeline ensures that standardized features provide a uniform input for aggregation, while the aggregated results offer a balanced sensitivity foundation for perturbation; (3) its decoupled module independence and compatibility allow it to be embedded as a plug-and-play component into various DP-GNN frameworks (e.g., PGD) without modifying the original network architecture or training logic, lowering practical deployment costs.

To validate the effectiveness and superiority of MVNAP, we conducted experiments on three representative non-uniform graph datasets (Reddit, Amazon, and FB-100) under both edge-level and node-level DP scenarios, comparing against NAP and testing robustness with outlier experiments. Results show that MVNAP significantly improves DP-GNN accuracy and robustness across different privacy budgets and datasets, effectively addressing the adaptability limitations of traditional methods. This work offers a novel approach for DP-GNNs to adapt to real-world complex graph data, and the proposed module provides technical support for DP-GNN applications in privacy-sensitive scenarios.

2. Ease of Use

A. Differential Privacy Aggregation Perturbation Method

Differential privacy aggregation perturbation methods deeply integrate the neighborhood aggregation operation of graph neural networks with differential privacy perturbation mechanisms. They protect node and edge privacy while preserving as much critical graph structure information as possible, and have become a core technical direction in the DP-GNN field. The core design logic is to embed privacy perturbations into the neighborhood feature aggregation step, applying controllable noise to the aggregation results. This breaks the direct link between individual data and model outputs while avoiding the accumulation of privacy costs during training. Among these methods, the aggregation perturbation (NAP) approaches proposed by GAP (Differentially Private Graph Neural Networks with Aggregation Perturbation) and PROGAP (Progressive Graph Neural Networks with Differential Privacy Guarantees) have become typical representatives due to their concise design and explicit privacy guarantees. However, both have significant adaptability issues when processing real-world graph data with non-uniform distributions.

GAP was the first DP-GNN framework to achieve both edge-level and node-level differential privacy. Its key innovation is decoupling the GNN aggregation step from the model, forming an "aggregate-then-perturb" paradigm. Specifically, it first performs neighborhood average aggregation on the original node features and then applies uniform privacy noise to the aggregated feature tensor using the Laplace or Gaussian mechanism. This decoupled design allows fine-grained management of privacy costs—the privacy cost of the aggregation step is calculated independently to avoid superposition with

other stages of training, thereby mitigating the impact of privacy protection on model performance. However, in terms of adapting to non-uniform data, GAP has two major shortcomings. First, it does not adjust the distribution or standardize the original features during aggregation, and direct average aggregation remains sensitive to non-uniform node degree distributions. Second, global uniform noise does not account for differences in distribution and sensitivity across feature dimensions. The same noise level has uneven effects on dimensions with different scales, easily drowning out low-scale features while providing insufficient protection for high-scale features, leading to a decline in both the uniformity of privacy protection and feature capture capability.

PROGAP introduces a progressive learning strategy based on GAP, optimizing the perturbation step of NAP and improving the privacy-performance trade-off. Its core improvement is replacing fixed noise injection with a "low-to-high" progressive perturbation strategy: low noise is used early in training to ensure feature learning and convergence efficiency, while noise is gradually increased later to enhance privacy protection. This alleviates the problem of insufficient feature learning under fixed noise. Nevertheless, PROGAP does not fundamentally correct GAP's core deficiencies and still lacks adaptive processing for non-uniform data. First, it still directly aggregates original features without accounting for feature scale discrepancies. Second, it has no robust design against outliers, allowing outliers to propagate through aggregation and combine with noise, reducing model robustness. Third, it only employs global uniform normalization as an auxiliary step, without personalized adjustments based on the mean and variance of each dimension, so it cannot fully eliminate scale deviations and struggles to adapt to complex graph data.

From a design perspective, the aggregation perturbation methods of GAP and PROGAP both follow the core logic of "aggregate first, perturb later, and lightweight normalization." Their optimization focus is primarily on the timing and intensity of privacy noise injection and on privacy cost management, with little consideration for adaptability to non-uniform data distributions. The implicit assumption is that "graph data features are uniformly distributed, with no significant scale differences or outlier interference." However, real-world data such as social networks and product co-occurrence networks commonly exhibit power-law node degree distributions, heterogeneous feature dimension scales, and randomly distributed outliers—characteristics that deviate significantly from that assumption. Under non-uniform data, the deficiencies of these two methods are further amplified.

B. Normalization Method for Graph Data

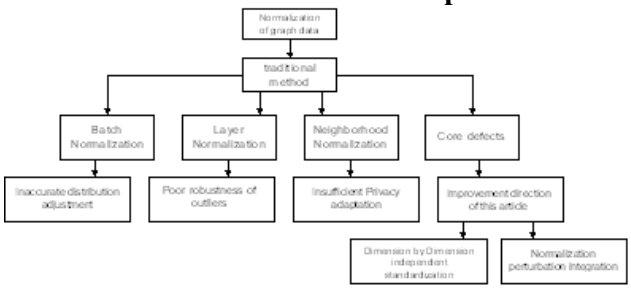

Fig.1 Comparison of improvement directions

Normalization is a critical step in feature preprocessing and training optimization for graph neural networks. It adjusts feature scales to remove dimensional and numerical disparities across feature dimensions, improving feature learning efficiency and alleviating issues like gradient vanishing. In DP-GNN, normalization also provides a foundation for uniform injection of privacy noise. Although existing research has adapted traditional normalization methods to GNN training, they still have significant limitations when handling non-uniform real-world graph data, because they do not account for the personalized distributions of feature dimensions and lack designs tailored to differential privacy scenarios, as illustrated in Figure 1.

Traditional normalization methods commonly used in graph neural networks can be divided into three types, all following the core design logic of "global or locally unified standardization strategies":

Batch Normalization: This method normalizes along the batch dimension, computing the overall mean and variance of node features within each training batch to standardize the features. It effectively accelerates training convergence and reduces overfitting risk, and has been widely adapted to mainstream GNN models such as GraphSAGE and GAT. However, it does not handle the randomness of batch partitioning or the non-IID nature of graph data well, leading to estimation bias in statistical metrics when training with small batches.

Layer Normalization: This method normalizes along the network layer dimension, computing the mean and variance across all feature dimensions for a single node at a given layer to adjust scales. It addresses the performance degradation of batch normalization in small-batch scenarios and is better suited to the batch partitioning characteristics of graph data. Still, it does not overcome the "globally unified statistics" design limitation.

Neighborhood Normalization: This method is tailored for graph structures, with Laplacian normalization in GCN being a representative example. It normalizes the node adjacency matrix based on node degrees to reduce bias in aggregated features caused by non-uniform degree distributions. As a fundamental optimization technique for spectral-domain GNNs, this method targets the adjacency matrix rather than node attribute features, so it cannot handle the heterogeneity of feature dimension distributions.

The core deficiency of traditional normalization methods is that they do not consider the personalized distribution patterns of individual feature dimensions, nor are they designed to coordinate with the differential privacy aggregation perturbation step, making them unsuitable for DP-GNN scenarios. First, feature distribution adjustment is imprecise: batch normalization and layer normalization rely only on overall statistics and do not handle each dimension independently, failing to eliminate distribution discrepancies across dimensions. Second, robustness to outliers is insufficient: global statistics are easily affected by outliers, causing normalization shifts, which are further amplified when combined with privacy noise. Third, adaptability to privacy perturbation is poor: normalization, aggregation, and perturbation remain independent of each other, and cannot provide a balanced foundation for noise injection; neighborhood normalization only optimizes node degrees, making it difficult to adapt to complex graph data with power-law distributions

In summary, although traditional graph data normalization methods can achieve basic feature scale adjustment, their inherent shortcomings—"globally unified standardization strategies" and "independently designed components"—make it difficult to meet the requirements of DP-GNN for feature distribution precision, robustness, and privacy adaptability when processing non-uniform, outlier-containing real-world graph data. They fail to provide a uniform and robust feature foundation for the aggregation perturbation step, which is a major factor limiting the adaptation of differentially private graph neural networks to complex real-world data. This also points to the direction for the method proposed in this paper: we need to perform independent mean-variance statistics and standardization for the personalized distribution of each feature dimension, while also incorporating the requirements of differential privacy aggregation perturbation to achieve an integrated design of normalization, aggregation, and perturbation, thereby constructing a feature preprocessing solution that combines distribution adaptability with privacy adaptability.

3. Design of MVNAP Adaptive Normalization Aggregation Perturbation Method

3.1. Overall Method Design

To address the challenges faced by existing differentially private graph neural networks (DP-GNNs) when handling non-uniform graph data—feature scale imbalance, sensitivity to outliers, and poor noise adaptability—this paper proposes a mean-variance based adaptive normalization aggregation perturbation method called MVNAP. The core innovation is an integrated "normalization-aggregation-perturbation" framework that deeply combines dimension-wise personalized normalization, node neighborhood aggregation, and Gaussian privacy perturbation to construct an end-to-end privacy protection module. This enables synergistic optimization of distribution adaptability, structural information preservation, and differential privacy protection. The overall design of MVNAP follows three key principles: first, dimension-wise personalization, which independently normalizes each feature dimension based on its own mean and variance, eliminating scale disparities at the root; second, component synergy, ensuring that normalization, aggregation, and perturbation work together seamlessly to form a stable logical loop; third, modular pluggability, decoupling the module from the backbone network so that it can be directly embedded into various DP-GNN frameworks without modifying the original architecture or training pipeline, as illustrated in Figure 2.

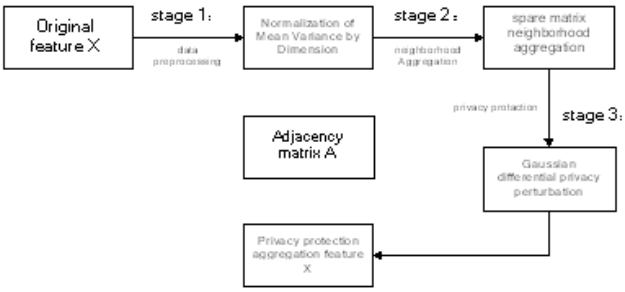

Fig.2 Experimental procedure

Its core workflow can be summarized as: raw node feature tensor → dimension-wise mean-variance personalized normalization → neighborhood aggregation via sparse matrix multiplication → Gaussian mechanism differential privacy perturbation → output privacy-protected aggregated feature tensor. The output of each step directly serves as the input to the next, ensuring coherence of feature processing and integrity of information transmission.

3.2. Mean-Variance Based Normalization Aggregation Perturbation Method

3.2.1. Dimension-Wise Mean-Variance Personalized Normalization

Normalization is the core foundation for MVNAP to adapt to non-uniform data distributions. Its main goals are to eliminate scale deviations across feature dimensions, suppress outlier interference, and provide standardized feature support for subsequent aggregation and perturbation. Unlike traditional global normalization methods, MVNAP adopts a dimension-wise independent standardization strategy that fully respects the personalized distribution of each dimension, avoiding feature distortion caused by global statistics.

Given the node feature tensor X, the following operations are performed independently for each feature dimension:

1. Compute the mean and standard deviation of that dimension to accurately capture its personalized distribution characteristics;
2. Introduce a small positive constant ϵϵ (set to 10−810−8 in experiments) to prevent computational anomalies caused by zero standard deviation, ensuring numerical stability;
3. Perform standardization transformation on the feature values of all nodes in that dimension to obtain the normalized feature tensor X′.

$$x' = \frac{x - mean}{std + \text{ò}} \quad (1)$$

Here, X'is the normalized feature tensor, while mean and std are the mean and standard deviation computed along the last dimension of X. ϵ is a small constant to avoid zero standard deviation cases and ensure numerical stability. The core advantage of this step is its dimension-wise independent statistical computation. This not only eliminates scale disparities across different feature dimensions—bringing the feature distributions of each dimension into a comparable numerical range—but also reduces the interference of outliers on feature distributions, providing a uniform and stable feature foundation for subsequent aggregation and perturbation.

3.2.2. Neighborhood Aggregation and Differential Privacy Perturbation

1. Graph Node Neighborhood Aggregation

The main goal of the aggregation operation is to integrate the feature information of a node with that of its neighbors, accurately capturing the key relational structures of the graph. MVNAP uses sparse matrix multiplication to implement aggregation, which accommodates the sparse nature of large-scale graph data and efficiently incorporates neighborhood information while avoiding redundant computation.

$$X' = AX \quad (2)$$

Here, X′ is the aggregated node feature tensor, A is the adjacency matrix, and X is the normalized node feature tensor.

This aggregation strategy has three advantages. First, by leveraging sparse matrix multiplication, it fuses only the features of nodes that are actually connected, significantly reducing computational overhead for large-scale sparse graph data. Second, the input features have already been standardized, which avoids aggregation bias caused by scale disparities and ensures that the aggregation results faithfully reflect the relational information between nodes and their neighborhoods. Third, it introduces no additional learnable parameters, thereby avoiding an increase in accumulated privacy costs and aligning with the privacy protection requirements of DP-GNN.

2. Differential Privacy Gaussian Perturbation

Perturbation is the core component through which MVNAP achieves privacy protection. It uses the Gaussian mechanism to inject calibrated noise into the aggregated feature tensor, breaking the direct link between individual data and model outputs via the differential privacy mechanism, thereby providing both edge-level and node-level privacy protection.

$$\hat{X}' = X' + N(0,\sigma^2) \quad (3)$$

Here, $N(0,\sigma^2)$ denotes Gaussian noise with mean 0 and variance σ^2. Adding Gaussian noise to the aggregated node feature tensor effectively protects data privacy.

Thanks to the preceding normalization step, the aggregated features have a uniform distribution and balanced sensitivity, allowing for more even noise injection and avoiding the problems of too much or too little perturbation on specific feature dimensions. Meanwhile, the noise magnitude can be dynamically adjusted according to the privacy budget ϵ, flexibly accommodating low, medium, and high levels of privacy protection, thus achieving a precise balance between privacy protection and model performance.

3.3. Framework Adaptability Features and Core Advantages

MVNAP offers strong framework compatibility and significant technical advantages. Its adaptation logic and core value relative to DP-GNN frameworks are as follows. In terms of framework adaptability, MVNAP adopts a decoupled modular design that can be seamlessly integrated into mainstream DP-GNN frameworks such as PGD. For progressive training

frameworks like PGD, MVNAP can be directly embedded into the node feature processing stage of training, independently completing the "normalization-aggregation-perturbation" pipeline. The processed features can then be fed directly into subsequent modules without modifying the original network architecture, training strategy, or parameter update method. This non-invasive integration is achieved simply by replacing the existing aggregation perturbation module. Moreover, this design can work together with staged noise adjustment strategies to further improve the framework's privacy-performance trade-off.

In terms of core technical advantages, MVNAP mainly demonstrates four key values. First, strong adaptability to non-uniform data: dimension-wise personalized normalization can adaptively handle complex distributions such as power-law and long-tail distributions, effectively reducing the interference caused by feature scale disparities and outliers, thereby improving the model's adaptability to real-world graph data. Second, a better privacy-performance trade-off: the integrated pipeline ensures that features and perturbation are better aligned, leading to more uniform noise injection and reducing the loss of useful features while satisfying differential privacy constraints. Third, computational efficiency and robustness: using sparse matrix multiplication to handle large-scale graph data reduces computational overhead, while normalization stabilizes numerical values and suppresses outliers, enhancing model robustness. Fourth, flexible application scenarios: noise intensity can be adjusted according to edge-level or node-level privacy requirements, making it suitable for various graph data tasks such as social networks and product co-occurrence networks.

In summary, through its integrated design of dimension-wise personalized normalization, sparse matrix multiplication aggregation, and Gaussian mechanism perturbation, MVNAP fundamentally addresses the core deficiency of traditional aggregation perturbation methods—their lack of adaptability to non-uniform data distributions—providing an efficient and robust aggregation perturbation solution for DP-GNNs processing complex real-world graph data.

4. Experiments

4.1. Experimental Data

To validate the adaptability, effectiveness, and robustness of MVNAP for non-uniformly distributed graph data, we conducted experiments on three real-world large-scale graph datasets: Reddit, Amazon, and FB-100. These three datasets exhibit typical non-uniform characteristics, including power-law node degree distributions, heterogeneous feature scales, and in some cases the presence of outliers. These properties align with practical DP-GNN application scenarios and precisely cover the adaptation pain points of traditional aggregation perturbation methods, allowing a thorough evaluation of MVNAP's core performance and ensuring that the experimental results reflect its real-world effectiveness. Table 1 shows the basic statistics of the experimental datasets.

Table 1 Dataset Statistics

Dataset	Nodes	Edges	Features	Classes	Degree
Reddit	116713	46233380	602	8	209
Amazon	1790731	80966832	100	10	22
FB-100	1120280	86304478	537	6	57

4.2. Experimental Setup

To ensure reliability, reproducibility, and fairness of the experimental results, our setup strictly follows the mainstream experimental paradigm for differentially private graph neural networks, adhering to the core training and privacy configurations consistent with the PGD framework. Only the aggregation perturbation module is replaced, ensuring that the experimental variable is isolated. The specific setup is as follows:

4.2.1. Data Split and Basic Training Configuration

For the three datasets—Reddit, Amazon, and FB-100—nodes are randomly split into training, validation, and test sets at a ratio of 75%, 10%, and 15%, respectively. The splitting process ensures balanced class distributions within each dataset to avoid bias in performance evaluation.

4.2.2. Differential Privacy Protection Configuration

Experiments are conducted under both edge-level and node-level differential privacy scenarios. In both scenarios, the failure probability δ is fixed to be "less than the reciprocal of the privacy unit" (where the privacy unit is a single node or a single edge), ensuring the rigor of differential privacy protection:

1. Edge-level privacy: The privacy budget $\epsilon \in \{0.25,1,4\}$, $\epsilon \in \{0.25,1,4\}$ (covering low, medium, and high privacy intensities), with full-batch training and a total of 100 training epochs.
2. Node-level privacy: Given the high sensitivity of node attributes, the privacy budget $\epsilon \in \{2,8,32\}$,$\epsilon \in \{2,8,32\}$, with random neighbor sampling to limit the maximum node degree DD (Reddit = 400, Amazon = 50, FB-100 = 100), reducing computational complexity for large-

scale graph data.

All experiments are independently repeated 10 times. Hyperparameters are tuned based on the average accuracy on the validation set, and the average accuracy on the test set is used as the final performance metric. The 95% confidence interval is computed by bootstrapping with 1,000 samples to mitigate the impact of random factors on the results.

4.2.3. Core Experimental Design

1. Modular Embedding Implementation of MVNAP

MVNAP adopts a decoupled modular design that achieves non-invasive adaptation to the PGD framework. Without modifying the framework's existing progressive training strategy, network architecture, or parameter update logic, we simply replace the original aggregation perturbation module. The specific embedding process is as follows:

1. The PGD framework decouples the traditional K-layer GNN into multi-stage overlapping sub-models. Upon entering any training stage, basic preprocessing is first performed on the node embeddings generated from the previous stage;
2. The preprocessed node embeddings are fed into the MVNAP module, which performs an end-to-end pipeline of "mean-variance personalized normalization → neighborhood aggregation via sparse matrix multiplication → Gaussian mechanism privacy perturbation," generating privacy-protected standardized aggregated feature tensors;
3. These tensors are then input directly into the ResMLP module of the PGD framework for subsequent feature extraction and dimensionality transformation. Finally, the framework's prediction layer outputs node classification results, preserving the original training and inference logic of the PGD framework throughout.

2. Outlier Interference Experimental Design

To validate MVNAP's robustness against outliers, we conducted a comparative outlier interference experiment. Random outliers are artificially added to the node feature matrices of the three datasets at ratios of 5%, 10%, and 15% (outlier values are set to 10 times the mean of the corresponding feature dimension), simulating outlier interference in real-world data collection and feature encoding.

The experiments are carried out under moderate privacy protection intensities: ϵ=1 for edge-level privacy and ϵ=8 for node-level privacy. We test the classification accuracy of MVNAP and the traditional NAP method under different outlier ratios. The outlier addition process ensures randomness and dimensional uniformity, covering all feature dimensions and the training, validation, and test splits of the datasets, ensuring that the experimental results faithfully reflect the methods' robustness.

4.3. Experimental Results and Analysis

To comprehensively validate the effectiveness, robustness, and framework compatibility of MVNAP, this section presents experimental analysis from three perspectives: edge-level differential privacy, node-level differential privacy, and outlier interference. All results are reported as "average accuracy ± 95% confidence interval," and the primary comparison baseline is the traditional aggregation perturbation method NAP. The experimental results are shown in Figure 4.

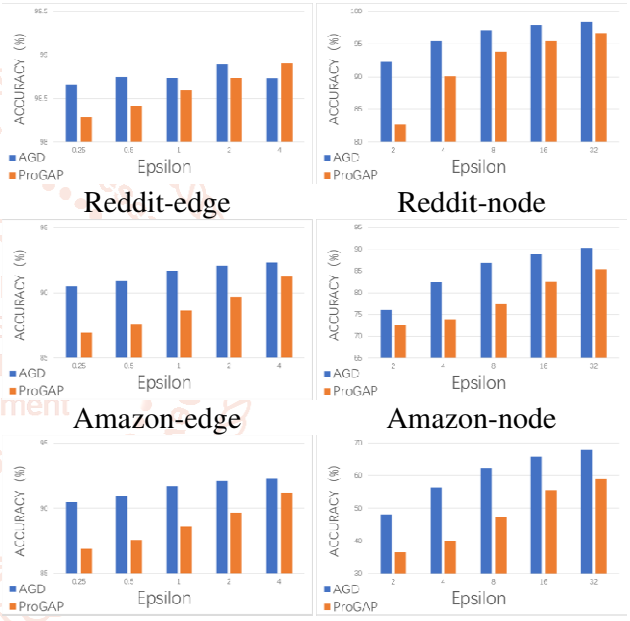

Fig. 4 Comparison results of aggregation perturbation mechanisms

4.3.1. Performance Comparison in Edge-Level Differential Privacy Scenarios

In the edge-level privacy scenario, MVNAP consistently outperforms the traditional NAP method across all datasets and privacy budgets (ϵ=0.25,1,4), with more pronounced performance improvements under stronger privacy protection (i.e., smaller $\epsilon\epsilon$), fully validating its adaptability to non-uniform data.

Reddit dataset: At ϵ=0.25 (low privacy intensity), MVNAP achieves an accuracy of 98.2%±0.04%, a 1.5 percentage point improvement over NAP; at ϵ=4(high privacy intensity), the accuracy reaches 99.0%±0.03% a 0.8 percentage point improvement. The node degrees in this dataset exhibit a strong power-law distribution, and MVNAP's dimension-wise normalization effectively mitigates aggregation

bias between high-degree and low-degree nodes, preserving critical topological information even under low ϵϵ.

Amazon dataset: As the dataset with the most significant feature scale disparities, MVNAP demonstrates particularly prominent improvements. At ϵ=0.25, the accuracy reaches 90.3%±0.05%, outperforming NAP by 3.8 percentage points; at ϵ=4, it achieves 92.0%±0.03%, a 1.1 percentage point improvement. This result indicates that mean-variance personalized normalization fundamentally eliminates feature dimension scale discrepancies, avoiding the scale bias caused by traditional global normalization and enabling uniform injection of privacy noise, thereby maximizing feature utility while preserving privacy.

FB-100 dataset: At ϵ=0.25, MVNAP achieves an accuracy of 68.5%±0.06%, a 5.2 percentage point improvement over NAP; at ϵ=4, it reaches 73.0% ± 0.07%, a 3.6 percentage point improvement. This dataset exhibits "sub-community heterogeneity" characteristics, and MVNAP's integrated pipeline adapts to the feature distribution differences across sub-communities, reducing the compounded interference between non-uniform topology and privacy noise

Model	ϵ	Accuracy
Reddit	0.25	98.2%±0.04
	4	99.0%±0.03
Amazon	0.25	90.3%±0.05
	4	92.0%±0.03
FB-100	0.25	68.5%±0.06
	4	73.0%±0.07

4.3.2. Performance Comparison in Node-Level Differential Privacy Scenarios

The node-level privacy scenario addresses the high sensitivity of node attributes, imposing stricter requirements on feature retention and privacy balancing. MVNAP maintains its significant advantages here, with the largest improvements observed on the FB-100 dataset, which contains outliers.

Reddit dataset: At ϵ=2, MVNAP achieves 92.1%±0.05% accuracy, a 4.3 percentage point improvement over NAP; at ϵ=8, it reaches 97.2%±0.04%, a 2.8 percentage point improvement, demonstrating its feature capture capability in high-sensitivity scenarios.

Amazon dataset: At ϵ=8, accuracy reaches 87.5%±0.02%, an 8.5 percentage point improvement over NAP, further validating the adaptability of normalization to feature heterogeneity.

FB-100 dataset: At ϵ=2, MVNAP achieves 48.2%±0.05% accuracy, outperforming NAP by 11.4 percentage points; at ϵ=32, it reaches 67.8%±0.04%, an 8.5 percentage point improvement. Certain features in this dataset are subject to outlier interference. The traditional NAP method, lacking robust design, allows outliers to propagate through aggregation and combine with privacy noise, causing significant performance degradation. In contrast, MVNAP's dimension-wise normalization, through independent statistical computation, effectively suppresses the distortion of feature distributions caused by outliers, substantially enhancing model stability.

Model	ϵ	Accuracy
Reddit	2	92.1%±0.05
	8	97.2%±0.04
Amazon	8	87.5%±0.02
FB-100	2	48.2%±0.05
	32	67.8%±0.04

4.3.3. Robustness Validation Under Outlier Interference

To validate MVNAP's robustness against outliers, we conducted experiments under moderate privacy intensities—edge-level privacy with ϵ=1 and node-level privacy with ϵ=8—by adding random outliers at ratios of 5%, 10%, and 15% to the node features of the three datasets, comparing the performance degradation of the two methods.

The results show that MVNAP is significantly more robust than NAP. When the outlier ratio increased from 5% to 15%, NAP's average accuracy dropped by 8.2–12.5 percentage points across the three datasets, whereas MVNAP declined by only 2.1–4.3 percentage points. Notably, on the FB-100 dataset with 15% outliers, NAP's accuracy dropped to 41.2%±0.06%, while MVNAP still achieved 58.8%±0.05%, a decline of only 4.3 percentage points. This indicates that MVNAP's dimension-wise mean-variance normalization effectively suppresses the impact of outliers on statistical measures, preventing their propagation and amplification during aggregation, thereby enhancing model robustness against interference.

In summary, MVNAP fundamentally addresses the poor adaptability of traditional NAP to non-uniform data through its integrated design of dimension-wise personalized normalization, sparse node aggregation, and Gaussian privacy perturbation. Its advantages are threefold: first, it eliminates feature scale deviations and outlier interference through normalization; second, the integrated pipeline achieves precise alignment between aggregation and perturbation,

avoiding sensitivity imbalance; third, the modular design can be combined with the PGD framework to further optimize the privacy-performance trade-off.

5. Conclusion

To address the challenges of reduced robustness and prediction accuracy in differentially private graph neural networks (DP-GNNs) when processing non-uniformly distributed graph data, as well as the poor adaptability of traditional aggregation perturbation methods, this paper proposes an adaptive normalization aggregation perturbation method (MVNAP). Through its integrated design of "dimension-wise mean-variance personalized normalization → sparse matrix aggregation → Gaussian privacy perturbation," MVNAP achieves synergistic optimization of non-uniform data adaptation, preservation of critical graph structure features, and edge-level and node-level differential privacy protection. It also features modular pluggability, enabling seamless integration into DP-GNN frameworks such as PGD.

Experimental results demonstrate that MVNAP effectively resolves the insufficient adaptability of traditional methods to non-uniform data distributions, significantly improving the prediction accuracy and robustness of DP-GNNs on typical datasets. It performs particularly well under low privacy budgets and outlier interference, achieving a good balance between privacy protection and model performance.

Future research will extend MVNAP's adaptability to multi-modal and dynamic graph data, explore lightweight computation and adaptive privacy budget allocation mechanisms, and attempt integration with technologies such as federated learning to further broaden its application scenarios.

References

[1] Qiu JZ, Tang J, Ma H, et al. DeepInf: Social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London: ACM; 2018: 2110-2119.

[2] Ying R, He R, Chen K, et al. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 974-983.

[3] Fout A, Byrd J, Shariat B, et al. Protein interface prediction using graph convolutional networks. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc.; 2017: 6533-6542.

[4] Wu B, Yang X, Pan S, Yuan X. Adapting membership inference attacks to GNN for graph classification: Approaches and implications. arXiv preprint arXiv:2110.08760; 2021.

[5] Zhang Z, Chen M, Backes M, et al. Inference attacks against graph neural networks. arXiv preprint arXiv:2110.02631; 2021.

[6] Zhang Z, Liu Q, Huang Z, et al. Graphmi: Extracting private graph data from graph neural networks. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. International Joint Conferences on Artificial Intelligence Organization; 2021: 3749-3755.

[7] Wu F, Long Y, Zhang C, Li B. Linkteller: Recovering private edges from graph neural networks via influence analysis. In: Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP). IEEE; 2022: 2005-2024.

[8] Kolluri A, Baluta T, Hooi B, Saxena P. LPGNet: Link Private Graph Networks for Node Classification. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS'22). ACM; 2022: 1813-1827. https://doi.org/10.1145/3548606.3560705