Theory and Practice of Science and Technology
2025, VOL. 6, NO. 6, 49-54
DOI: 10.47297/taposatWSP2633-456910.20250606

A Survey of Privacy-Preserving Techniques in Federated Graph Neural Networks

Lingyu Wang

School of Arts and Sciences, China University of Petroleum-Beijing at Karamay, Karamay, Xinjiang, 834000, China

ABSTRACT
Federated Graph Neural Networks (FedGNNs) are increasingly adopted in high-sensitivity domains—such as finance, healthcare, and government—imposing stringent requirements on privacy, performance, and communication efficiency. This survey systematically reviews four core privacy-preserving techniques (homomorphic encryption, differential privacy, secure multi-party computation, and trusted execution environments) and introduces an evaluation framework that balances privacy, performance, and efficiency. Within this framework, we compare encryption-based aggregation, embedding perturbation, graph-structure anonymization, uncertainty-aware distillation, and decentralized optimization. To handle dynamic graphs and multimodal data, we propose a sliding-window temporal encoding and cross-modal alignment strategy, and design a spatiotemporal multimodal defense pipeline with adaptive aggregation and privacy-budget scheduling for robust, efficient protection. Finally, we identify future challenges—standardized evaluation, adversarial-defense benchmarks, and industrial deployment—and outline design principles to guide secure FedGNN implementations in sensitive environments.

KEYWORDS
Federated learning; Graph neural networks; Privacy preservation; Differ- ential privacy; Homomorphic encryption; Decentralization; Multimodal fusion

1 Introduction and technical foundations

1.1 Graph neural networks: architectures and privacy constraints

Graph Neural Networks (GNNs) model structured data by iteratively aggregating neighbor information into context-aware node embeddings. Among the most influential designs, the Graph Convolutional Network (GCN) employs Laplacian-based convolutions in a lightweight, efficient form but requires full adjacency visibility. The Graph Attention Network (GAT) assigns learned attention weights to neighbors, better capturing structural heterogeneity at the expense of greater computation and communication. GraphSAGE adopts an inductive sampling-and-aggregation paradigm that scales across graphs and underpins many federated frameworks. Despite their expressiveness and scalability, these architectures depend on centralized data access—raising privacy risks under regulations such as the GDPR and China's Personal Information Protection Law.

1.2 Federated learning: paradigms and architectures

Federated Learning (FL) enables multiple clients to jointly train a global model via local computation and parameter exchange, without sharing raw data. In horizontal FL, clients share the same feature space but hold disjoint samples; vertical FL handles overlapping samples with complementary feature sets, requiring secure feature-alignment protocols; federated transfer learning suits scenarios where both samples and features partially overlap, emphasizing cross-domain adaptability. Architecturally, FL can be centralized, with a server orchestrating aggregation—simplifying deployment but creating a single point of failure—or decentralized, where clients synchronize peer-to-peer, enhancing robustness and privacy resilience.

1.3 Emergence of federated graph neural networks

Federated Graph Neural Networks (FedGNNs) merge FL and GNNs to learn graph representations across institutions—such as banks, healthcare providers, and social platforms—without exposing raw topology or node attributes. To overcome structural invisibility, FedGNNs employ edge-prediction or pseudo-neighbor expansion to infer connectivity under privacy constraints. They establish shared embedding spaces for cross-client representation alignment, supporting both global consistency and client-specific personalization. At the system level, cryptographic key exchanges and decentralized synchronization protocols reduce reliance on a central server and mitigate leakage risks. By extending federated learning to graph-structured data, FedGNNs open new avenues for multi-party collaborative intelligence; yet, balancing communication overhead, model accuracy, and privacy guarantees remains an open challenge.

2 Privacy-preserving strategies and core mechanisms in FEDGNN

Graph Neural Networks (GNNs) power applications in recommendation, finance, and healthcare, demanding both efficiency and privacy. Federated GNNs (FedGNNs) meet this need by training across clients without sharing raw data, yet risks like structural leakage, embedding inference, and communication attacks remain. This section reviews five key FedGNN privacy strategies—encrypted computation, differential privacy, structural anonymization, uncertainty modeling, and decentralized optimization—highlighting their principles, representative models, challenges, and real-world applicability.

2.1 Encrypted communication and secure aggregation Mech-anisms

FedGNNs exchange gradients, parameters, and embeddings, which, if sent in plaintext, invite model inversion or membership inference. Three main encryption-based aggregation strategies have been proposed:

(1) Homomorphic Encryption (HE): VFGNN encrypts updates so the server aggregates ciphertexts directly, fully protecting graph structure at the cost of ×3 computation and ×2 bandwidth.

(2) P2P Masked Synchronization: D-FedGNN uses the Diffie – Hellman key exchange for peer-to-peer masking and unmasking of local updates, improving robustness but introducing protocol complexity that scales linearly with participant count.

(3) Transport-Layer Security: FedGCN and FedPerGNN employ TLS/SSL handshakes and certificate verification to secure all exchanges, incurring under 5% slowdown in convergence while integrating seamlessly into existing FL frameworks.

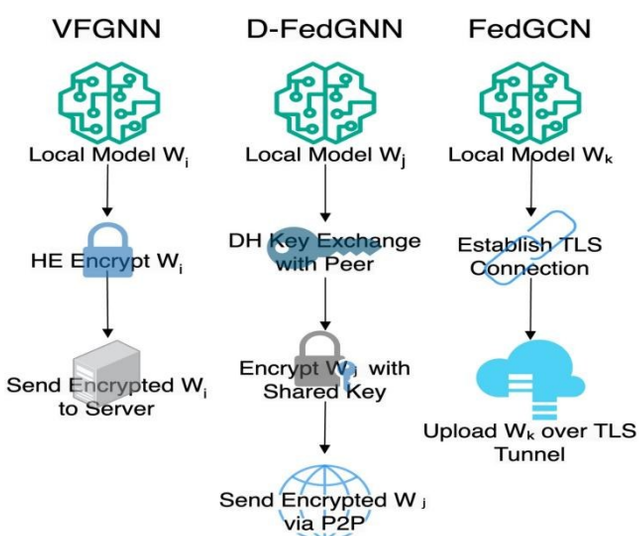

Figure 1 Illustration of three typical encrypted communication mechanisms in FedGNN: VFGNN (HE aggregation), D-FedGNN (DH-based masked synchronization), and FedGCN (TLS-secured transmission).

These methods trade off privacy guarantees, computational and communication costs, and deployment complexity. Future work could adopt hybrid encryption—using homomorphic encryption only for the most sensitive embeddings while securing other updates with symmetric ciphers or TLS—leverage Trusted Execution Environments to speed up ciphertext operations, and incorporate real-time privacy risk assessment to adjust encryption schemes and communication intervals dynamically, thus balancing performance and privacy across diverse applications.

2.2 Differential privacy and embedding perturbation strategies

FedGNNs enforce Differential Privacy by first clipping the L_2 norm of local gradients or node embeddings to a threshold C, then adding Gaussian noise sampled from $N(0,\sigma^2C^2)$, and finally performing secure aggregation of the perturbed updates to form the global model. The noise multiplier σ is calibrated based on the privacy budget (ε,δ), the number of communication rounds T, and C, while accounting methods such as Rényi DP or zCDP track cumulative privacy loss across iterations.

To defend further against inversion and membership inference attacks, some FedGNN variants train generative models (e. g., VAEs or GANs) on the real embedding distribution to synthesize pseudo-interactions. By mixing these synthetic samples with genuine updates, the true embedding patterns are obfuscated, significantly enhancing privacy protection.

Recent works have significantly enhanced DP ' s utility in FedGNNs. Chen et al. integrate variational Bayesian inference

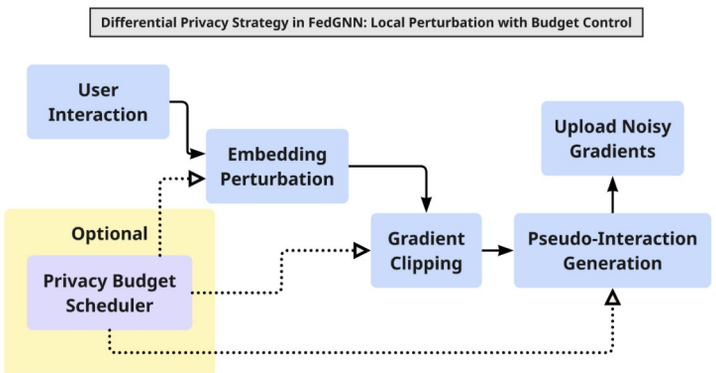

Figure 2 Illustration of the differential privacy process in FedGNN: embeddings or gradients are first clipped and then perturbed with Gaussian noise. An optional privacy scheduler dynamically adjusts \(\sigma\) during training. A pseudo-interaction generator may further confuse adversaries by introducing synthetic samples.

with DP noise injection to learn the noise distribution, mitigating performance degradation and boosting node-classification accuracy. Wang et al. combine multi-domain learning with adaptive noise, tuning perturbation levels to client heterogeneity and improving generalization under non-IID data. Despite DP's theoretical guarantees, real-world deployment demands carefully balancing the privacy budget, noise magnitude, and model utility. Future research should explore adaptive, task-aware clipping thresholds, dynamic privacy-budget allocation, hybrid encryption schemes, and unified frameworks that blend adversarial training with explainability analysis to further optimize the privacy–utility trade-off.

2.3 Graph structure obfuscation and topological anonymization mechanisms

In federated graph learning, structural information—particularly edge connections—can be exploited by adversaries to infer sensitive node associations. Graph structure obfuscation thus injects controlled perturbations into local subgraphs to blur the true topology and curb structural leakage.

Zhang et al. In FedPerGNN augment each node's true neighbor set with pseudo-neighbors and their features before aggregation, using a weighted scheme that preserves semantic similarity while confusing attackers. FedSage+ instead randomly inserts extra edges during graph construction, then applies a learnable edge-weight mechanism to offset the perturbations' impact on non-IID data distributions. Later, FedRA integrates an adversarial discriminator to adaptively adjust pseudo-edge generation based on the loss between real and perturbed structures, further improving the privacy–utility trade-off.

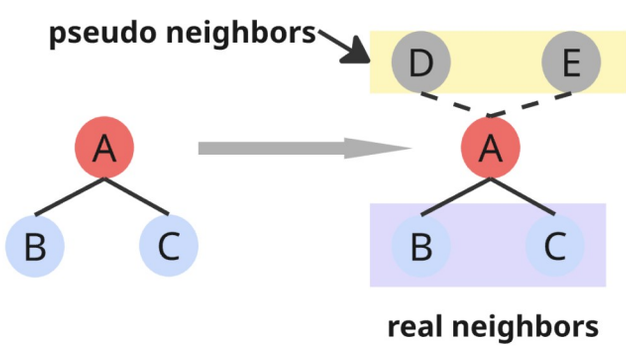

Figure 3 Illustration of structure obfuscation: pseudo-neighbors D and E are injected in addition to real neighbors B and C for node A. Learnable edge weights are assigned to hinder adversarial reconstruction of the true topology.

These obfuscation strategies inevitably trade anonymity for stability: overly aggressive perturbations can disrupt model convergence, while too light an intervention may leave higher-order structural inference attacks unchecked.

2.4 Bayesian modeling and uncertainty-aware integration

BI-FedGNN addresses data heterogeneity and local noise by embedding variational Bayesian inference into each client's training. Instead of sending raw gradients, a client first fits a prior over its GNN parameters and computes a variational posterior, yielding predictive-distribution parameters (mean and variance). A private distillation module then draws representative samples—optionally perturbed to satisfy (\varepsilon,\delta)-DP—and uploads either these samples

or their summary statistics for aggregation, thereby reducing communication overhead and ensuring compatibility with differential privacy.

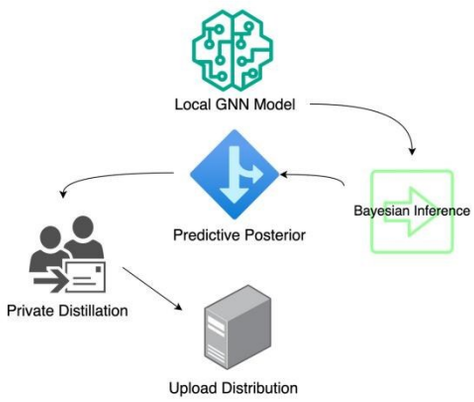

Figure 4 Bayesian uncertainty modeling and private distillation in BI-FedGNN: local prior specification → variational posterior → sample generation → optional perturbation → upload for aggregation.

Empirically, BI-FedGNN outperforms standard FedGNNs by 2–5% on public node-classification benchmarks and cuts communication overhead by ~40%, although its variational-inference and distillation steps raise computational demands, challenging deployment on resource-limited edge devices.

2.5 Decentralized training architectures and optimization strategies

Traditional FedGNNs depend on a central server for global aggregation, creating a single point of failure and exposing the server to privacy risks. D-FedGNN overcomes this by adopting peer-to-peer masked synchronization: clients exchange encrypted, masked updates directly, then collaboratively unmask and aggregate them—eliminating central coordination while preserving privacy.

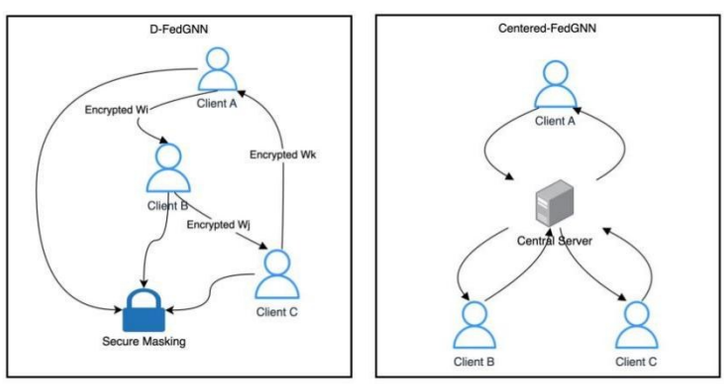

Figure 5 Comparison of centralized and decentralized FedGNN architectures. Left: P2P masked synchronization in D-FedGNN; Right: traditional server-based centralized aggregation.

However, the resulting many-to-many communication pattern inflates bandwidth demand and makes unmasking sensitive to network delays, introducing synchronization bias. To mitigate these challenges, recent proposals include hierarchical aggregation—grouping clients by subgraph similarity for staged intra- and inter-group synchronization; asynchronous updates—allowing clients to push local updates on demand with delay-compensation or sliding-window mechanisms to reduce staleness; and secure compression—applying quantization or sparse encoding atop masking to shrink message size without sacrificing privacy.

2.6 Trends in multi-mechanism integration

Recent FedGNN research has shifted from single defenses to layered frameworks that blend complementary techniques. For example, MDD-FedGNN fuses differential privacy with adversarial discrimination and adaptive noise, BI-FedGNN unifies variational Bayesian inference with DP-driven distillation, and FedRA combines DP, structural perturbation, and knowledge distillation to handle heterogeneous data. Tables 1 and 2 summarize these multi-mechanism designs.

Table 1

Model	Structure	DP	Encryption	Decentral.	Distillation
FedGNN	No	Yes	No	No	No
FedPerGNN	Yes	No	Yes	No	No
BI-FedGNN	No	Yes	No	No	Yes
D-FedGNN	No	No	Yes	Yes	No
MDD-FedGNN	Yes	Yes	No	No	Yes

Table 2

Model	Privacy Mechanism	Key Features
FedGNN	Local DP + embedding perturbation	Lightweight local perturbation to protect embeddings
FedPerGNN	Pseudo-neighbors + SSL	Topology obfuscation via graph anonymization
BI-FedGNN	Bayesian inference + DP	Uncertainty modeling and distributional distillation
D-FedGNN	DH encryption + secure masking	Peer-to-peer secure synchronization
MDD-FedGNN	DP + adversarial discriminator	Adaptive noise and distillation under non-IID threats

3 Privacy–performance–efficiency trade-offs

3.1 Privacy–performance–efficiency tension

In sensitive domains such as finance, healthcare and public administration, FedGNNs must simultaneously safeguard data privacy, maintain model accuracy and minimize communication overhead. Noise injection or structural obfuscation —integral to privacy—inevitably distorts gradients or graph topologies, degrading node - classification accuracy and slowing convergence, which in turn increases communication rounds. Conversely, constrained bandwidth and latency limit the extent of privacy-enhancing transformations that can be applied. Thus, privacy, performance and efficiency form a tightly coupled loop in which gains along one axis impose costs on the others.

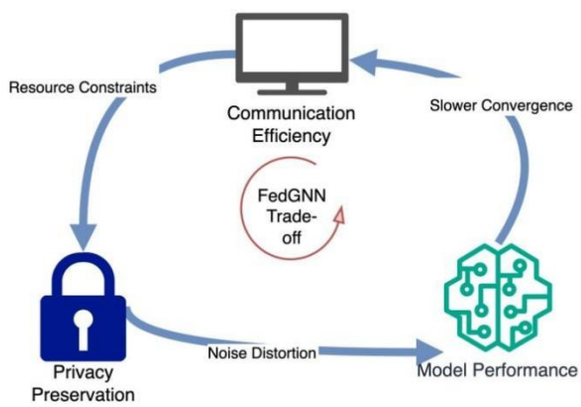

Figure 6 Illustration of the privacy–performance–efficiency trade-off loop in FedGNN

3.2 Heterogeneity and trust challenges

Client graphs often differ markedly in density, scale and feature distributions—consider a dense social-media network versus a sparse financial transaction graph—and these disparities destabilize global aggregation methods like FedAvg, leading to fluctuating accuracy and erratic convergence rates. Personalized embedding layers or subgraph clustering can mitigate divergence on small scales, but their benefits taper off in large, highly heterogeneous deployments and demand extensive hyperparameter tuning. Meanwhile, adversarial threats—malicious node injections, structural perturbations and backdoor attacks—exploit unmasked updates to compromise model integrity, and the lack of standardized robustness benchmarks further undermines trust in federated settings.

3.3 Communication bottlenecks and scalability strategies

Deep GNN architectures and large - scale graphs generate hundreds of megabytes per training round. Even with parameter quantization (reducing size by ~90 %), hierarchical aggregation or asynchronous updates, network constraints can introduce prohibitive latency, impeding timely model updates. Hybrid communication schemes that preserve high-fidelity transmission for critical subgraphs while aggressively compressing less impactful regions, along with adaptive switching between centralized and peer - to - peer modes based on real - time bandwidth predictions, offer paths to scalability without sacrificing convergence.

4 Toward scalable and trustworthy FEDGNNS

4.1 Adaptive privacy–utility management

Dynamic privacy-budget scheduling—loosening constraints during early structure learning and tightening them as training stabilizes—combined with Bayesian uncertainty modeling or selective encryption, can reconcile formal privacy guarantees with high accuracy.

4.2 Heterogeneity-aware and robust architectures

Developing explainable client-similarity metrics, lightweight subgraph alignment techniques and meta-learning aggregation rules will enable stable training across evolving, non-IID data distributions. Integrating runtime anomaly detection, client trust scoring and decentralized masking into the training pipeline, together with federated-GNN adversarial-defense benchmarks, will enhance system resilience.

4.3 Dynamic modeling and standardized evaluation

Extending FedGNNs to dynamic, multimodal scenarios through temporal attention, sliding-window encodings and cross-modal alignment is crucial for real-world applications. Finally, establishing a unified evaluation framework—with clear leakage metrics (e. g., membership-inference success), privacy-grading protocols and benchmark datasets—alongside regulatory and semantic alignment guidelines, will underpin the scalable, trustworthy deployment of FedGNN solutions.

About the Author

Lingyu Wang, wangling@st.cupk.edu.cn

References

[1] Zhang, Z., & Li, H. (2024). A comprehensive survey on trustworthy graph neural networks.
[2] Li, W., & Wang, M. (2024). A survey on federated graph neural networks: Methods and applications.
[3] Chen, Y., & Zhao, L. (2024). FedPerGNN: Privacy-preserving personalized recommendation via federated graph neural networks.
[4] Wu, X., & Zhou, Y. (2023). Vertically federated graph neural network for privacy-preserving node classification.
[5] Chen, Q., & Sun, R. (2023). BI-FedGNN: Bayesian inference based federated GNN with privacy-aware uncertainty quantification.
[6] Wang, L., & Liu, Y. (2023). MDD-Fed: Multi-defense distillation based federated GNN against backdoor attack.
[7] Liu, F., & Zhang, L. (2023). A survey on privacy-preserving techniques in federated graph neural networks.
[8] Zhao, M., & Yang, C. (2023). A survey of privacy protection technologies in federated learning.
[9] Zhang, H., & Zhou, Q. (2022). FedPerGNN: A privacy-preserving personalized federated graph neural network framework.
[10] Zhu, L., & Wang, X. (2023). Isomorphic federated graph neural network for privacy-preserved supply chain data sharing.
[11] Cheng, F. (2024). Research and application of federated graph neural network models.
[12] Sun, M., & Liu, J. (2023). Isomorphic federated graph neural network.
[13] He, K., & Zhang, L. (2022). Federated graph neural network for privacy-preserving recommendation.