

PROVABLY COMMUNICATION-EFFICIENT FEDERATED GRAPH NEURAL NETWORK

Anonymous authors

Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) are powerful tools for relational data, but their application is often limited by data silos and privacy concerns, as real-world graphs are frequently distributed across multiple clients. While federated learning (FL) offers a privacy-preserving training paradigm, existing federated GNN approaches suffer from a critical flaw: they either ignore the crucial links between clients, sacrificing accuracy, or require impractically high communication overhead. We introduce CE-FedGNN, a communication-efficient federated GNN framework for such coupled graphs. Instead of sharing raw data or per-iteration embeddings, CE-FedGNN infrequently transmits only aggregated, high-level embeddings, preserving critical structural context while minimizing privacy leakage and communication costs. Despite the challenges of optimization under multi-layer composition and coupled data, we establish a convergence rate of $O(1/\sqrt{T})$ to a stationary point while the communication complexity is $O(T^{3/4})$. We further derive bounds for injecting Gaussian noise that provide formal differential privacy. Our experiments on a synthetic interbank anti-money laundering task show that the effectiveness of CE-FedGNN, which can be preserved even with injected Gaussian noise for differential privacy.

1 INTRODUCTION

Graph neural networks (GNNs) have become a leading paradigm for learning from relational data, achieving state-of-the-art results in areas such as social network analysis, traffic forecasting, and financial fraud detection (Shu et al., 2019; Derrow-Pinion et al., 2021; Egressy et al., 2024). By propagating and aggregating information through graph structures, GNNs capture complex dependencies and emergent patterns that are invisible to traditional methods. However, most existing approaches assume centralized access to the entire graph, which is an assumption that is rarely valid in practice due to strict privacy regulations (e.g., GDPR, CCPA) and competitive business barriers. For example, in cross-bank money laundering detection, financial institutions each observe only a fragment of the global transaction graph and cannot share raw data with others.

Federated learning (FL) provides a natural framework for collaborative model training without centralizing sensitive data. Standard FL methods such as FedAvg (Konečný et al., 2016; McMahan et al., 2017) are designed for independent local datasets, where the global objective is simply the average loss across clients. Graph data violates this assumption: edges connect nodes across clients, and a node’s neighborhood often spans multiple institutions. This cross-client coupling makes neighborhood aggregation, which is a core GNN operation, dependent on external information. Ignoring cross-client edges leads to incomplete and inaccurate models, while simply sharing the necessary embeddings at every iteration introduces prohibitive communication costs. Moreover, the multi-layer compositional structure of GNNs amplifies noise and bias from partial information, further complicating optimization. These challenges are especially acute in applications such as financial crime detection, where malicious behaviors such as laundering cycles or “smurfing” schemes are emergent properties of the global graph. No single bank can detect such patterns in isolation, underscoring the need for communication-efficient, privacy-preserving methods that faithfully capture cross-institutional dependencies. An illustration has been presented in Figure 1(a).

Existing federated GNN methods present a stark trade-off between modeling accuracy and communication efficiency. One line of work simplifies the problem by entirely ignoring cross-client edges

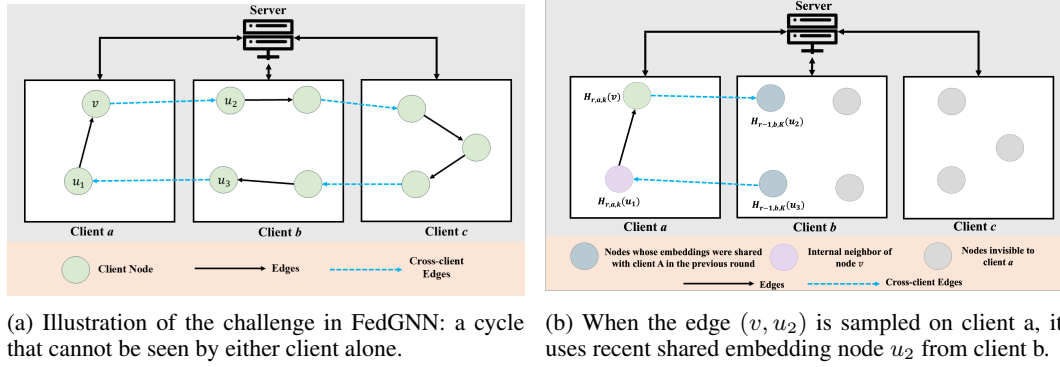


Figure 1: Illustration of the challenge in FedGNN and our algorithmic design.

(Du et al., 2022; Peng et al., 2022; Scardapane et al., 2020; He et al., 2021; Pan et al., 2023; Hu et al., 2022), sacrificing the model’s ability to capture critical global graph properties. Conversely, methods that do account for these edges (Chen et al., 2021; Wu et al., 2022; 2021) typically require the costly exchange of node embeddings in every training iteration. A middle-ground approach shares embeddings only once at initialization (Yao et al., 2023), but this fails to adapt to representation drift during training, leading to suboptimal accuracy. Sharing embeddings only once at initialization fails to track representation drift and degrades accuracy. (Du & Wu, 2022) reduces communication by infrequently sharing node embeddings. However, this approach overlooks nodes not sampled in the most recent global round and requires feature sharing, which raises privacy concerns. Moreover, its theoretical guarantees only ensure convergence to a neighborhood of the solution. Another method (Qiu et al.) uses global connections infrequently but relies primarily on local graphs, which may still obscure important global patterns. (Guo et al., 2023) addresses cross-client coupling in federated compositional optimization via decomposing the gradient into active/passive parts, where active parts depend on local data and passive parts depend on other clients. The summary statistics are shared every K steps. While provably communication-efficient, their framework cannot apply to GNNs, which have a multilayer compositional structure and interdependence between inner data and outer. What is more, the above methods lack a formal privacy guarantee. In federated GNNs, sharing aggregated embeddings is less sensitive than sharing individual embeddings Yao et al. (2023), however, recent work has shown that sharing intermediate embeddings in GNNs can still leak sensitive information (Zhang et al., 2024; Duddu et al., 2020; Zhang et al., 2022; Li et al., 2020).

To this end, we design a communication-efficient FedGNN algorithm and analyze the communication and iteration complexity with and without differential privacy. We study a realistic setting where an edge between two clients is visible to both parties (i.e., edge attributes are shared), while node attributes remain private. For example, when a transaction occurs between two banks, both banks observe its attributes (e.g., time, amount), but each retains only its own node features. Our contributions are as follows.

- **Decomposition framework for federated GNNs.** We design a tailored decomposition scheme that maintains a moving-average estimator of node embeddings to mitigate the variance and bias from mini-batch updates. Crucially, this mechanism is applied only to nodes (not edges), as edges can vastly outnumber nodes in practice. When a neighbor lies on another client, we use its most recently shared moving-average embedding. We prove this provides sufficiently accurate estimates by explicitly accounting for the resulting latency error. This allows us to restrict cross-client interactions to 1-hop neighbors and avoid multi-hop sampling. The framework, illustrated in Figure 1(b), further reduces privacy risks by transmitting only high-level embeddings instead of intermediate representations from every layer. We also extend the moving-average strategy to gradients. With T iterations, our algorithm achieves a convergence rate of $O(1/\sqrt{T})$ toward a stationary point while requiring only $O(T^{3/4})$ communication rounds.
- **Differential privacy mechanisms.** In addition to conventional noise injection into model parameters and gradients, we introduce Gaussian noise into shared embeddings. We analyze the convergence rate under the injected noise.

- **Empirical evaluation.** Experiments on a synthetic anti-money laundering task demonstrate the effectiveness and communication efficiency of our method. Furthermore, with Gaussian noise injection, we provide differential privacy, and our experiments confirm that strong utility is retained even under significant noise levels.

2 RELATED WORK

Graph Neural Networks. Unlike images or text, many real-world problems involve entities connected by complex relationships that cannot be naturally represented in Euclidean space. Graph neural networks (GNNs) provide a principled way to handle such data by propagating and aggregating information across graph structures. They have shown remarkable success in domains such as social networks (Badrinath et al., 2025; Hou et al., 2025), transportation systems (Derrow-Pinion et al., 2021; Zheng et al., 2023), simulation (Sanchez-Gonzalez et al., 2020; Jain et al., 2025), and biological systems (Bongini et al., 2021). Beyond these, GNNs are particularly advantageous for reasoning tasks that require capturing structural dependencies, including combinatorial optimization (Dudzik & Veličković, 2022; Bevilacqua et al., 2023) and financial fraud detection (Egressy et al., 2024; Lin et al., 2024), highlighting scenarios where expressive GNN architectures are particularly crucial. More discussion of recent advances of GNN can be found in Appendix A.

Communication-Efficient Federated Learning. A primary challenge in Federated Learning (FL) is communication overhead. A vast body of work addresses this for problems where the global objective is a separable average of individual client losses (Konečný et al., 2016; McMahan et al., 2017; Stich, 2018; Yu et al., 2019a;b; Yang, 2013; Karimireddy et al., 2020; Kairouz et al., 2021; Khaled et al., 2020; Woodworth et al., 2020b;a; Haddadpour et al., 2019; Deng & Mahdavi, 2021; Deng et al., 2020; Liu et al., 2020; Sharma et al., 2022; Li et al., 2022; Huang et al., 2022; Tarzanagh et al., 2022; Xing et al., 2022).

However, these conventional FL algorithms struggle with coupled objectives where client data is interdependent. Some works address specific forms of coupling: Yuan et al. (2021); Guo et al. (2020) study AUC maximization by reformulating it into a decomposable minimax problem, though this approach lacks generality. Gao et al. (2022) provides convergence analysis for compositional problems but ignores cross-client coupling. Guo et al. (2023) propose a communication-efficient method for a general pairwise objective using an active-passive gradient decomposition. While a step forward, their two-level compositional setup assumes inner and outer data are independent, making it inapplicable to the complex dependencies in federated GNNs. Overall, many existing solutions are largely ad hoc and lack strong theoretical foundations (Han et al., 2022; Zhang et al., 2020; Wu et al., 2022; Li & Huang, 2022).

Particularly for federated GNNs, which are subject to the challenge of cross-client edges, most existing approaches either ignore these edges, (Du et al., 2022; Peng et al., 2022; Scardapane et al., 2020; He et al., 2021). or incur prohibitive communication costs by requiring clients to share node embeddings every training iteration (Wu et al., 2022; 2021). Some works attempt to reduce this cost by heuristics utilizing global information sparsely Yao et al. (2023); Qiu et al..

Privacy in Federated Learning. While the above methods on federated compositional problems aim to reduce communication overhead, they largely overlook the privacy implications of sharing high-dimensional embeddings. Differential privacy (DP) has become the dominant formal approach for limiting information leakage in FL (Abadi et al., 2016; McMahan et al., 2018; Truex et al., 2020; Wei et al., 2020). In federated GNNs, very few formally analyze privacy guarantee, while intermediate representations can encode sensitive information about nodes and their neighborhoods, especially in financial or social domains.

3 METHOD

We now formalize the problem, introduce our communication-efficient federated GNN algorithm, and extend it with Gaussian perturbations to ensure differential privacy.

3.1 PROBLEM STATEMENT

A large class of modern GNNs is formulated under the Message Passing framework (Wu et al., 2020). The central mechanism is to iteratively aggregate information from neighbors and update node representations. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathbb{E})$ with node set $\mathcal{V}$ and edge set $\mathbb{E}$. In each layer l , a node v aggregates messages from its neighbors $\mathcal{N}(v)$ and updates its representation:

$$\mathbf{h}^{(l)}(v) = \text{UPDATE}^{(l)}\left(\mathbf{h}^{(l-1)}(v), \text{AGGREGATE}^{(l)}(\{\mathbf{h}^{(l-1)}(u) : u \in \mathcal{N}(v)\})\right) \quad (1)$$

where the AGGREGATE function is typically permutation-invariant (e.g., summation or mean).

In this work, we focus on a common setup in which the aggregation function is chosen from GCN (Kipf & Welling, 2017), GraphSAGE-mean (Hamilton et al., 2017), or GIN (Xu et al., 2018), and the update function sums the node’s previous embedding with the aggregated message, followed by a learnable transformation (e.g., a linear layer and nonlinearity). Without loss of generality, we adopt GraphSAGE-mean, which can be written as:

$$\mathbf{h}^{(l)}(v) = \phi\left(\mathbf{W}^{(l)} \cdot \frac{1}{|\mathcal{N}(v) \cup v|} \sum_{u \in \mathcal{N}(v) \cup v} \mathbf{h}^{(l-1)}(u)\right), \quad (2)$$

where ϕ denotes a nonlinear activation function. We let $\hat{\mathbf{h}}^l(v) := \sum_{u \in \mathcal{N}(v) \cup v} \mathbf{h}^{(l-1)}(u)$, $\tilde{\mathbf{h}}^l(v) := \mathbf{W}^{(l)} \hat{\mathbf{h}}^l(v)$ and thus $\mathbf{h}^{(l)}(v) = \phi(\tilde{\mathbf{h}}^l(v))$. We simplify the notation $\mathcal{N}(v) \cup v$ to $\mathcal{N}(v)$ when the context is clear. The formulations of GCN and GIN are provided in Appendix A.

For node- or edge-level prediction, a linear layer is applied to the final embeddings:

$$\hat{y}_x = F(\mathbf{W}^{L+1}; \mathbf{h}(x)), \quad \mathcal{L}(x) = \ell(\hat{y}_x, y_x), \quad x \in \mathcal{X}, \quad (3)$$

where $\mathcal{X} = \mathcal{V}$ for the node classification task and $\mathcal{X} = \mathcal{E}$ for the edge classification task, and edge representations are further computed via mean aggregation of their endpoints. For an edge $e = (u, v)$, we have

$$\mathbf{h}(e) = \phi\left(\mathbf{W}^e \cdot \frac{\mathbf{h}^{(L)}(u) + \mathbf{h}^{(L)}(v)}{2}\right). \quad (4)$$

In federated learning, with $\mathbf{W}$ denoting the model of all layers, the formulation is

$$F(\mathbf{W}) = \frac{1}{N} \sum_{i=1}^N F_i(\mathbf{W}), \quad \text{where} \quad F_i(\mathbf{W}) = \frac{1}{|\mathcal{X}_i|} \sum_{x \in \mathcal{X}_i} \mathcal{L}(x). \quad (5)$$

3.2 A COMMUNICATION-EFFICIENT ALGORITHM

To illustrate the challenge, consider computing gradients on a given client. For instance, the gradient of $F_i(\mathbf{W})$ with respect to $\mathbf{W}^e$ for an edge $e = (u, v)$, where node v lies outside client i , is

$$\frac{\partial F_i(\mathbf{W}; e)}{\partial \mathbf{W}^e} = \left(\frac{\partial F_i(\mathbf{W}; e)}{\partial \mathbf{h}(e)} \right) \phi' \left(\mathbf{W}^e \cdot \frac{\mathbf{h}^{(L)}(u) + \mathbf{h}^{(L)}(v)}{2} \right) \frac{\mathbf{h}^{(L)}(u) + \mathbf{h}^{(L)}(v)}{2}. \quad (6)$$

The highlighted term depends on the embedding of a cross-client neighbor. This leads to several challenges: 1) **Cross-client dependency**. Local clients cannot compute the neighbor embedding $\mathbf{h}^{(L)}(v)$ on their own; 2) **High communication cost**. Naively exchanging embeddings at every iteration incurs prohibitive communication overhead; 3) **Gradient backpropagation**. By the chain rule, gradients must also flow through the cross-client embedding $\mathbf{h}^{(L)}(v)$, which further complicates local updates. 4) **Biased gradients**. Even with embedding exchange, estimating $\mathbf{h}^{(l-1)}(u)$ from minibatches produces bias due to the nonlinear composition of $\phi(\cdot)$ and $\phi'(\cdot)$.

We propose to maintain *moving averages* of both (i) intermediate embeddings and (ii) gradient estimators, which provide low-variance approximations. Let subscript r, i, k denote round r on client i at iteration k . For clarity, we first assume that a node’s neighbors are restricted to those residing in the same client, while cross-client interactions occur only through inter-client edges. (In Appendix F, we extend the discussion to cases where neighbors of nodes may span multiple clients.)

The forward pass for a node v with a moving-average update is defined as

$$\begin{aligned}\tilde{H}_{r,i,k}^{(l)}(v) &= (1 - \gamma)\tilde{H}_{r,i,k-1}^{(l)}(v) + \gamma \frac{1}{n_{r,i,k}(v)} \mathbf{W}_{r,i,k-1}^{(l)} \cdot \sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k}^{(l-1)}(u), \\ H_{r,i,k}^{(l)}(v) &= \phi(\tilde{H}_{r,i,k}^{(l)}(v)),\end{aligned}\quad (7)$$

where $n_{r,i,k}(v) = |\mathcal{N}_{r,i,k}(v)|$, $\mathcal{N}_{r,i,k}(v)$ are local neighbors in the batch, and the input embeddings satisfy $H^{(0)}(\cdot) = h^{(0)}(\cdot)$. We further denote $\mathbf{W}_{r,k} := \frac{1}{N} \sum_i \mathbf{W}_{r,i,k}$.

The behavior of this estimator is captured by the following lemma:

Lemma 3.1. *Under appropriate conditions, with p being the smallest probability for a node to be sampled, the Algorithm 1 ensures that*

$$\frac{1}{RK} \sum_R \sum_k \|H_{r,i,k}^{(l)}(u) - h^{(l)}(u)\|^2 \leq O\left(\frac{1}{\gamma p RK} + \frac{\gamma}{p} + \gamma^2 K^2 + \frac{\|\mathbf{W}_{r,k} - \mathbf{W}_{r,k-1}\|^2}{\gamma^2}\right). \quad (8)$$

Remark. By appropriately choosing parameters γ and K , the first three error terms decrease with the number of rounds R . To control the final term, we next introduce a gradient estimator that ensures model updates are computed from low-variance estimates. Consequently, the variance of the embedding estimator H converges along with the gradient estimator and true gradient. Importantly, only 1-hop node embeddings need to be communicated across clients, since each node maintains a moving average of its embeddings. This design eliminates the need for multi-hop cross-client sampling.

The batch stochastic gradient over W_e is

$$\hat{\nabla} F_i(\mathbf{W}_{r,i,k}^e; B_{r,i,k}) = \left(\frac{\partial F_i(\mathbf{W}_{r,i,k}^e; e; B_{r,i,k})}{\partial h(e)} \phi' \left(\mathbf{W}_{r,i,k}^e \cdot \hat{h}_{r,i,k}(e) \right) \hat{h}_{r,i,k}(e) \right). \quad (9)$$

where

$$\hat{h}_{r,i,k}(e) = \frac{\mathbf{h}_{r,i,k}^{(L)}(u) + \mathbf{h}_{r-1,c(v),K}^{(L)}(v)}{2}, \quad (10)$$

with $c(v)$ denoting the host client of node v . Here, $B_{r,i,k}$ denotes a minibatch sampled on client i , and the subscript H of F indicates that moving-average estimators are applied in the forward pass. The key difficulty is that the gradient depends on late-layer embeddings from other clients (highlighted in blue). By the chain rule, backpropagation requires differentiating through this term, which a local client cannot perform.

We resolve this issue by observing that the edge $e = (u, v)$ also exists in client $c(v)$. Without loss of generality, we assume e is sampled with equal probability on both $c(u)$ and $c(v)$; otherwise, one can reweight the sampling or explicitly instruct $c(v)$ to sample e in the next round. The gradient decomposition on client $c(v)$ is symmetric, therefore, client $c(v)$ can compute gradients through the blue term, while client $c(u)$ only needs to provide embeddings of u . As a result, although each client computes only part of the gradient, their aggregated contributions, $\frac{1}{N} \sum_i \hat{\nabla} F_i(\mathbf{W}_{r,i,k}^e; B_{r,i,k})$, recover the full global gradient.

Then we apply a moving average for the gradient estimation:

$$G_{r,i,k}^{(l)} = (1 - \beta)G_{r,i,k}^{(l)} + \beta \hat{\nabla} F_i(\mathbf{W}; B_{r,i,k}), \quad (11)$$

whose behavior is analyzed in the following lemma:

Lemma 3.2. *Under appropriate conditions, with $\bar{G}_{r,k} = \frac{1}{N} \sum_i G_{r,i,k}$ the Algorithm 1 ensures that*

$$\mathbb{E} \|\bar{G}_{r,k} - \nabla F(\mathbf{W}_{r,k})\|^2 \leq O\left(\frac{1}{\beta RK} + \frac{1}{\gamma p RK} + \frac{\beta}{N} + \frac{\gamma}{p} + \beta^2 K^2 + \frac{\eta^2}{\beta^2} \|\nabla F(\mathbf{W}_{r,k-1})\|^2\right).$$

Algorithm 1 CE-FedGNN**On Server:**Initialization: $\mathbf{W}^{(l)}, \forall l$, global buffer of embeddings $\mathcal{H}$ **for** $r \in [1, 2, \dots, R]$ **do**

Collect requests of neighbor embeddings from all clients, and route them to corresponding clients, collect those embeddings from those banks, and send them to clients.

Clients in parallel do:

 $\mathbf{W}_{r,i}, \mathcal{H}_{r,i} \leftarrow \text{LocalUpdate}(\mathbf{w}_{r,i})$ $\mathbf{W}_r = \frac{1}{N} \sum_{i \in [N]} \mathbf{W}_{r,i,K}, G_r^{(l)} = \frac{1}{N} \sum_{i \in [N]} G_{r,i,K}^{(l)}$ $\mathbf{W}_{r,i,0} = \mathbf{W}_r, G_{r,i,0} = G_r$ Flush $\mathcal{H}$ with $\mathcal{H}_{r,i}$ **end for**Output: $\mathbf{w}^{(E+1,0)}$.**On Local Client $\mathbf{W}$:**Initialization: Receive $\mathbf{W}_{r,i,0}^l$ from the server. Received updated node embeddings from neighbors.**for** $k \in [K]$ **do**Locally sample a mini-batch $\mathcal{B}$ as 0-hop seeds, then sample L -hop.

Update edge and node embeddings using (40)

Update gradient estimator using (11)

Update model:

$$\mathbf{W}_{r,i,k} = \mathbf{W}_{r,i,k-1} - \eta G_{r,i,k}. \quad (12)$$

end forReturn $\mathbf{w}_{r,i,k}$, and updated margin node embeddings $\mathcal{H}_{r,i}$ to the server.

Remark. With proper parameter setting, the estimator G becomes accurate as the model converges. Finally, the theorem of overall convergence analysis is

Theorem 3.3. *Under appropriate conditions, the Algorithm 1 ensures that*

$$\frac{1}{R} \sum_r \|\nabla F(\mathbf{W}_{r-1})\|^2 \leq O\left(\frac{1}{\eta RK} + \frac{1}{\beta RK} + \frac{1}{\gamma p RK} + \frac{\beta}{N} + \frac{\gamma}{p} + \beta^2 K^2\right). \quad (13)$$

Remark. By setting $\gamma = O(\frac{1}{R^{2/3}})$, $\beta = O(\frac{1}{R^{2/3}})$, $\eta = O(\frac{1}{R^{2/3}})$ and $K = O(R^{1/3})$, we have $\frac{1}{R} \sum_r \|\nabla F(\mathbf{W}_{r-1})\|^2 \leq \frac{1}{R^{2/3}}$, i.e. $\frac{1}{R} \sum_r \|\nabla F(\mathbf{W}_{r-1})\|^2 \leq \frac{1}{T^{1/2}}$, where $T = RK$ and $R = T^{3/4}$. To get $\mathbb{E}[\frac{1}{R} \sum_{r=1}^R \|\nabla F(\bar{\mathbf{w}}^r)\|^2] \leq \alpha^2$, we just set $R = O(\frac{1}{\alpha^3})$, $\eta = O(\alpha^2)$, $\gamma = O(\alpha^2)$, $\beta = O(\alpha^2)$ and $K = \frac{1}{\alpha}$ to yield iteration complexity of $O(1/\alpha^4)$ and communication complexity of $O(1/\alpha^3)$. We also observe that more clients can speed up the convergence by reducing the term $\frac{\beta}{N}$ and also by increasing p compared to a centralized setting. In the best case, it could provide a linear speed-up, in which case by setting $R = O(\frac{1}{\alpha^3})$, $\eta = O(N\alpha^2)$, $\gamma = O(N\alpha^2)$, $\beta = O(N\alpha^2)$ and $K = \frac{1}{N\alpha}$, the total iteration complexity becomes $O(1/N\alpha^4)$.

At the end of each round r , clients communicate their model parameters $\mathbf{W}_{r,i,K}$, gradient estimator $G_{r,i,K}$ and update embeddings H of boundary node (i.e., those needed by other clients) to neighboring clients connected via edges. The overall procedure is summarized in Algorithm 1. For intermediate embeddings $\tilde{H}$, each client transmits only the updated embeddings of its local nodes, yielding communication cost $O(Kd)$, where K is the number of local steps and d is the embedding dimension. It is noticeable that the number of shared nodes is typically far smaller than the number of edges. If not updated, stale embeddings from previous rounds are reused, but moving averages ensure they are accurate in expectation. Unlike (Guo et al., 2023), our method requires no auxiliary data for shared statistics; this efficiency stems from the GNN sampling strategy and the fact that each node maintains its own moving average of embeddings.

3.3 DIFFERENTIAL PRIVACY EXTENSION AND ANALYSIS

We now extend our algorithm to incorporate differential privacy (DP) and analyze its convergence properties. While DP for shared models/gradients has been well studied in standard federated learning, existing analyses do not account for the compositional dependencies unique to GNNs. In our setting, the local updates of H and G remain unchanged, but Gaussian noise is added to H , G , and $\mathbf{w}$ whenever they are communicated across clients.

Specifically, when an external neighbor is required, the update of H on a client is perturbed by Gaussian noise, and the forward pass of an edge is modified if one of its endpoints belongs to another client:

$$\hat{\nabla} F_i(\mathbf{W}_{r,i,k}^e; B_{r,i,k}) = \left(\frac{\partial F_i(\mathbf{W}_{r,i,k}^e; e; B_{r,i,k})}{\partial h(e)} \phi' \left(\mathbf{W}_{r,i,k}^e \cdot \hat{h}_{r,i,k}(e) \right) \hat{h}_{r,i,k}(e) \right). \quad (14)$$

where

$$\hat{h}_{r,i,k}(e) = \frac{\mathbf{h}_{r,i,k}^{(L)}(u) + \mathbf{h}_{r-1,c(v),K}^{(L)}(v) + \mathcal{G}(0, \sigma_0^2 I)}{2} \quad (15)$$

Local updates of $\mathbf{W}$ and G remain the same as in the previous subsection, but during communication rounds, Gaussian noise is injected into their aggregated forms:

$$\mathbf{W}_{r+1,i,0} = \frac{1}{N} \sum_{i=1}^N \mathbf{W}_{r,i,K} + \mathcal{G}(0, \sigma_1^2 I), \quad (16)$$

$$G_{r+1,i,0} = \frac{1}{N} \sum_{i=1}^N G_{r,i,K} + \mathcal{G}(0, \sigma_2^2 I). \quad (17)$$

Their behavior can be bounded by the following lemmas, respectively.

Lemma 3.4. *Under appropriate conditions, DP version of Algorithm 1 ensures that*

$$\|H_{r,i,k}^{(l)}(e) - h^{(l)}(e)\|^2 \leq O\left(\frac{1}{p\gamma RK} + \gamma + \beta^2 K^2 + \frac{\|\mathbf{W}_{r,k}^{(l)} - \mathbf{W}_{r,k-1}^{(l)}\|^2}{\gamma} + \sigma_0^2\right). \quad (18)$$

Lemma 3.5. *Under appropriate conditions, DP version of Algorithm 1 ensures that*

$$\left\| \frac{1}{N} \sum_i G_{r,i,k} - \nabla F(\mathbf{W}) \right\|^2 \leq O\left(\frac{1}{\gamma RK} + \gamma + \beta^2 K^2 + \frac{\|\mathbf{W}_{r,k}^{(l)} - \mathbf{W}_{r,k-1}^{(l)}\|^2}{\gamma} + \sigma_0^2 + \sigma_1^2 + \sigma_1/\beta\right).$$

The overall convergence behavior is

Theorem 3.6. *Under appropriate conditions, by setting $\eta = \Theta(\beta) = \Theta(\gamma)$, DP version of Algorithm 1 ensures*

$$\frac{1}{R} \sum_r \|\nabla F(\mathbf{W}_{r-1})\|^2 \leq O\left(\frac{1}{\eta RK} + \beta + \beta^2 K^2 + \sigma_0^2 + \sigma_1^2 + \sigma_1/\beta + \sigma_2^2 + \sigma_2/\beta\right). \quad (19)$$

Remark. By (McMahan et al., 2018; Abadi et al., 2016), to ensure (ϵ, δ) -differential privacy, the added noise should be $\propto \frac{1}{M}$, where M is the number of contributing data entries. For model updates and gradient estimator, all nodes and edges in the batch are possible to contribute, therefore, only a small noise needs to be added, but for the shared node embedding, only neighboring nodes can contribute, thus σ_0 should be much bigger than σ_1 and σ_2 . Fortunately, the convergence has a much better dependence on σ_0 compared to σ_1 and σ_2 .

4 EXPERIMENTS

We conduct experiments to assess the effectiveness, utility under differential privacy, and communication efficiency of the proposed algorithm.

Data. Due to regulatory constraints, real-world anti-money laundering (AML) transaction data are not publicly available. We therefore adopt the realistic simulator from Altman et al. (2023), which generates financial transaction networks by modeling agents such as banks and individuals, while injecting illicit activity through well-established laundering patterns. The reliance on synthetic data underscores the need for studying federated learning in this domain, where direct access to financial data is inherently limited.

We evaluate on two small, two medium, and two large datasets, each provided in two variants: a high illicit ratio (HI) and a low illicit ratio (LI). For example, *HI-Small* refers to the small-scale dataset with a high fraction of illicit transactions. Following Altman et al. (2023), we use a temporal train-validation-test split by transaction timestamp. To simulate federated deployment, data are partitioned across 4–32 clients. For cross-client edges, each client retains a copy of the corresponding edge features, reflecting the realistic setting of inter-bank transactions. Detailed dataset statistics are reported in Appendix D.

Setup. While the proposed method can be applied to message-passing GNNs with different aggregation functions as discussed in Section 3.1, in the experiments we focus on GIN (Xu et al., 2018) (a message-passing GNN with provably maximum expressive power) and PNA (Veličković et al.) (employing multiple aggregators in parallel), both augmented with edge updates, reverse messaging, Ego ID, and port numbering as described by Egressy et al. (2024). Each model consists of two message-passing layers followed by a classification head.

We compare against the following baselines: 1) Single Client (SC) that trains on each client independently, 2) FedAVG (McMahan et al., 2017) that only communicates the model weights, 3) Swift-FedGNN (Qiu et al.) that trains on global graph infrequently while only using local graphs mostly, and 4) FedGCN (Hu et al., 2022) which shares node embedding only once at the beginning.

Hyperparameters are kept consistent across methods: local seed batch size is 1k for HI datasets and 2k for LI datasets; hop-1 and hop-2 neighbor sampling sizes are both set to 100; local update steps per round are fixed to 32, unless otherwise specified. All algorithm are run for 20k iterations. Performance is measured by average F1 score across all clients.

Primary results. Tables 1 and 2 summarize results on HI and LI datasets. On the high-illicit (HI) datasets, our method consistently outperforms all baselines across dataset scales. In particular, Ours-PNA achieves the best overall performance—for example, on HI-Large, it improves F1 from 0.6235 (FedAVG-GIN) to 0.7114.

All methods experience performance degradation under the low-illicit (LI) setting, as expected due to stronger class imbalance. Nevertheless, our method maintains a clear advantage: Ours-PNA achieves 0.3158 F1 on LI-Large, significantly outperforming alternatives. These results suggest our approach is more robust to class imbalance, likely due to its ability to exploit structural patterns by sharing node embeddings even when illicit transactions are rare.

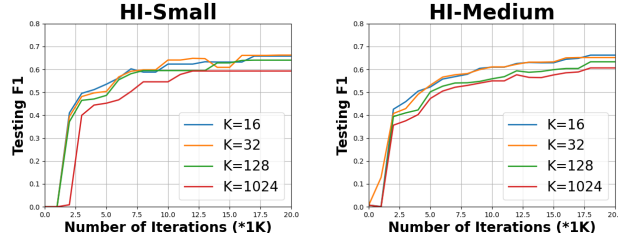
Table 1: Results for High Illicit Ratio Datasets.

	HI-Small	HI-Medium	HI-Large
SC-GIN	0.1526 $\pm$ 0.0157	0.3572 $\pm$ 0.0305	0.2416 $\pm$ 0.0266
SC-PNA	0.4409 $\pm$ 0.0294	0.5305 $\pm$ 0.0311	0.5744 $\pm$ 0.0182
FedAVG-GIN	0.4103 $\pm$ 0.0335	0.5421 $\pm$ 0.0273	0.6235 $\pm$ 0.0310
Swift-GIN	0.3873 $\pm$ 0.0306	0.5689 $\pm$ 0.0282	0.6339 $\pm$ 0.2205
FedGCN-GIN	0.4152 $\pm$ 0.0291	0.5538 $\pm$ 0.0310	0.5817 $\pm$ 0.0338
FedAVG-PNA	0.5427 $\pm$ 0.0288	0.5037 $\pm$ 0.0341	0.5958 $\pm$ 0.0299
Swift-PNA	0.5746 $\pm$ 0.0302	0.5722 $\pm$ 0.0258	0.6144 $\pm$ 0.0372
FedGCN-PNA	0.5653 $\pm$ 0.0365	0.6292 $\pm$ 0.0283	0.6028 $\pm$ 0.0324
CE-FedGNN-GIN	0.4916 $\pm$ 0.0218	0.6024 $\pm$ 0.0314	0.6461 $\pm$ 0.0306
CE-FedGNN-PNA	0.6623 $\pm$ 0.0273	0.6517 $\pm$ 0.0322	0.7114 $\pm$ 0.0251

Communication Efficiency We show results of varying communication interval K for training CE-FedGNN-PNA in Figure 2. It is clear that skipping a lot of communication does not degrade the performance, even with $K = 1024$, our method still maintains a great advantage over other baselines shown in Table 1.

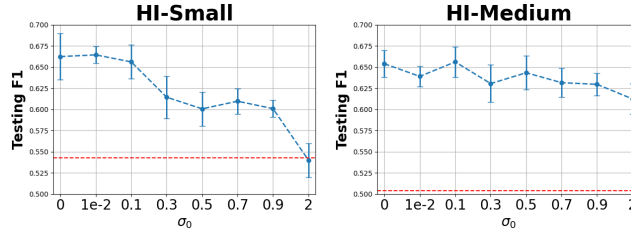
Table 2: Results for Low Illicit Ratio Datasets.

	LI-Small	LI-Medium	LI-Large
SC-GIN	0.1238 ± 0.0101	0.0746 ± 0.0153	0.0080 ± 0.0143
SC-PNA	0.1396 ± 0.0148	0.2472 ± 0.0200	0.1262 ± 0.0198
FedAVG-GIN	0.0000 ± 0.0000	0.0068 ± 0.0012	0.0000 ± 0.0000
Swift-GIN	0.0000 ± 0.0000	0.0061 ± 0.0016	0.1294 ± 0.0246
FedGCN-GIN	0.1702 ± 0.0316	0.0647 ± 0.0083	0.0000 ± 0.0000
FedAVG-PNA	0.1556 ± 0.0131	0.2614 ± 0.0225	0.0000 ± 0.0000
Swift-PNA	0.1995 ± 0.0206	0.2001 ± 0.0280	0.0000 ± 0.0000
FedGCN-PNA	0.1726 ± 0.0239	0.1821 ± 0.0197	0.0000 ± 0.0000
CE-FedGNN-GIN	0.1630 ± 0.0155	0.0828 ± 0.0127	0.1917 ± 0.0147
CE-FedGNN-PNA	0.2655 ± 0.0199	0.2918 ± 0.0106	0.3158 ± 0.0215

Figure 2: Ablation Study: Varying K

Differential Privacy Finally, we study the effect of differential privacy by adding Gaussian noise $\mathcal{N}(0, \sigma^2 I)$. Since the effect of noise on model/gradient are well studied, we focus on the noise on the shared embeddings, by fixing $\sigma_1 = \sigma_2 = 3 \times 10^{-3}$ and vary σ_0 . We bound the Euclidean norm of embedding to 10, and set $K = 1024$ to reduce sharing frequency. The (ϵ, δ) coefficient of differential privacy is computed using moments accountant of (Abadi et al., 2016; McMahan et al., 2018). Details are discussed in Appendix C. Results are shown in Figure 3.

We observe a clear privacy-utility tradeoff: moderate noise levels have little effect on performance, but as σ_0 increases, F1 scores degrade significantly. Interestingly, our method remains competitive with FedAVG even under stronger noise, indicating better resilience to DP perturbations.

Figure 3: Performance under different σ_0 . The red dashed line represents the performance of FedAVG without injected Gaussian noise.

5 ETHICS STATEMENT

This work introduces CE-FedGNN, a communication-efficient federated graph neural network framework for collaborative learning across distributed and privacy-sensitive graph data. All experiments are conducted on synthetic datasets simulating financial transaction networks; no proprietary, personal, or confidential real-world data were used. Our method is designed to reduce privacy risks by limiting cross-client data exchange and incorporating formal differential privacy guarantees. We acknowledge that real-world deployment in domains such as financial crime detection raises broader ethical considerations, including fairness, transparency, and potential misuse. We emphasize that further interdisciplinary research and stakeholder oversight are required before applying these techniques in operational financial or regulatory systems.

REFERENCES

- Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.
- Erik Altman, Jovan Blanuša, Luc Von Niederhäusern, Béni Egressy, Andreea Anghel, and Kubilay Atasü. Realistic synthetic financial transactions for anti-money laundering models. *Advances in Neural Information Processing Systems*, 36:29851–29874, 2023.
- Anirudhan Badrinath, Alex Yang, Kousik Rajesh, Prabhat Agarwal, Jaewon Yang, Haoyu Chen, Jiajing Xu, and Charles Rosenberg. Omniverse: Large scale, multi-entity heterogeneous graph representation learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 4261–4272, 2025.
- Beatrice Bevilacqua, Kyriacos Nikiforou, Borja Ibarz, Ioana Bica, Michela Paganini, Charles Blundell, Jovana Mitrovic, and Petar Veličković. Neural algorithmic reasoning with causal regularisation. In *International Conference on Machine Learning*, pp. 2272–2288. PMLR, 2023.
- Pietro Bongini, Monica Bianchini, and Franco Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450:242–252, 2021.
- Mário Cardoso, Pedro Saleiro, and Pedro Bizarro. Laundrograph: Self-supervised graph representation learning for anti-money laundering. In *Proceedings of the third ACM international conference on AI in finance*, pp. 130–138, 2022.
- Fahao Chen, Peng Li, Toshiaki Miyazaki, and Celimuge Wu. Fedgraph: Federated graph learning with intelligent sampling. *IEEE Transactions on Parallel and Distributed Systems*, 33(8):1775–1786, 2021.
- Yuyang Deng and Mehrdad Mahdavi. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency. In *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1395. PMLR, 2021.
- Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33:15111–15122, 2020.
- Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3767–3776, 2021.
- Bingqian Du and Chuan Wu. Federated graph learning with periodic neighbour sampling. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*, pp. 1–10. IEEE, 2022.
- Hanbiao Du, Meng Shen, Rungeng Sun, Jizhe Jia, Liehuang Zhu, and Yanlong Zhai. Malicious transaction identification in digital currency via federated graph deep learning. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6. IEEE, 2022.
- Vasisht Duddu, Antoine Boutet, and Virat Shejwalkar. Quantifying privacy leakage in graph embedding. In *MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pp. 76–85, 2020.
- Andrew J Dudzik and Petar Veličković. Graph neural networks are dynamic programmers. *Advances in neural information processing systems*, 35:20635–20647, 2022.
- Béni Egressy, Luc Von Niederhäusern, Jovan Blanuša, Erik Altman, Roger Wattenhofer, and Kubilay Atasü. Provably powerful graph neural networks for directed multigraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11838–11846, 2024.
- Hongchang Gao, Junyi Li, and Heng Huang. On the convergence of local stochastic compositional gradient descent with momentum. In *International Conference on Machine Learning*, pp. 7017–7035. PMLR, 2022.

- Zhishuai Guo, Mingrui Liu, Zhuoning Yuan, Li Shen, Wei Liu, and Tianbao Yang. Communication-efficient distributed stochastic auc maximization with deep neural networks. In *International Conference on Machine Learning*, pp. 3864–3874. PMLR, 2020.
- Zhishuai Guo, Rong Jin, Jiebo Luo, and Tianbao Yang. Fedxl: provable federated learning for deep x-risk optimization. In *International Conference on Machine Learning*, pp. 11934–11966. PMLR, 2023.
- Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *Advances in Neural Information Processing Systems*, 32, 2019.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation, 2022. URL <https://arxiv.org/abs/2207.09158>.
- Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning system and benchmark for graph neural networks. *arXiv preprint arXiv:2104.07145*, 2021.
- Junpeng Hou, Wei-Ting Lin, Arkin Dharawat, Jiaying Qu, Qi Wang, Sai Xiao, Xianxing Zhang, and Weiran Li. Optimize visual shopping journey with embedding-based retrieval in pinterest closeup. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4329–4334, 2025.
- Kai Hu, Jiasheng Wu, Yaogen Li, Meixia Lu, Liguang Weng, and Min Xia. Fedgcn: Federated learning-based graph convolutional networks for non-euclidean spatial data. *Mathematics*, 10(6):1000, 2022.
- Yankun Huang, Qihang Lin, Nick Street, and Stephen Baek. Federated learning on adaptively weighted nodes by bilevel optimization. *arXiv preprint arXiv:2207.10751*, 2022.
- Ayush Jain, Ehsan Haghighat, and Sai Nelaturi. Latticegraphnet: A two-scale graph neural operator for simulating lattice structures. *Engineering with Computers*, 41(2):829–844, 2025.
- Guillaume Jaume, An-phi Nguyen, María Rodríguez Martínez, Jean-Philippe Thiran, and Maria Gabrani. edgcn: a simple and powerful gnn for directed labeled graphs. *arXiv preprint arXiv:1904.08745*, 2019.
- 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.
- Hiroki Kanezashi, Toyotaro Suzumura, Xin Liu, and Takahiro Hirofuchi. Ethereum fraud detection with heterogeneous graph neural networks. *arXiv preprint arXiv:2203.12363*, 2022.
- 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*, pp. 5132–5143. PMLR, 2020.
- Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020.
- Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- Jakub Konečný, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

- Junyi Li and Heng Huang. Fedgrec: Federated graph recommender system with lazy update of latent embeddings. *arXiv preprint arXiv:2210.13686*, 2022.
- Junyi Li, Jian Pei, and Heng Huang. Communication-efficient robust federated learning with noisy labels. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 914–924, 2022.
- Kaiyang Li, Guangchun Luo, Yang Ye, Wei Li, Shihao Ji, and Zhipeng Cai. Adversarial privacy-preserving graph embedding against inference attack. *IEEE Internet of Things Journal*, 8(8): 6904–6915, 2020.
- Junhong Lin, Xiaojie Guo, Yada Zhu, Samuel Mitchell, Erik Altman, and Julian Shun. Fraudgt: a simple, effective, and efficient graph transformer for financial fraud detection. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pp. 292–300, 2024.
- Mingrui Liu, Wei Zhang, Youssef Mroueh, Xiaodong Cui, Jarret Ross, Tianbao Yang, and Payel Das. A decentralized parallel algorithm for training generative adversarial nets. *Advances in Neural Information Processing Systems*, 33:11056–11070, 2020.
- 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*, pp. 1273–1282. PMLR, 2017.
- H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.
- Jack Nicholls, Aditya Kuppa, and Nhien-An Le-Khac. Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape. *Ieee Access*, 9:163965–163986, 2021.
- Qiyang Pan, Ruofan Wu, Tengfei Liu, Tianyi Zhang, Yifei Zhu, and Weiqiang Wang. Fedgkd: Unleashing the power of collaboration in federated graph neural networks. *arXiv preprint arXiv:2309.09517*, 2023.
- Huailiang Peng, Yujun Zhang, Hao Sun, Xu Bai, Yangyang Li, and Shuhai Wang. Domain-aware federated social bot detection with multi-relational graph neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.
- Peiwen Qiu, Ziyue Luo, Jia Liu, and Ness Shroff. Swift-fedgmn: Federated graph learning with low communication and sample complexities.
- Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html>.
- Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Approximation ratios of graph neural networks for combinatorial problems. *Advances in Neural Information Processing Systems*, 32, 2019.
- Simone Scardapane, Indro Spinelli, and Paolo Di Lorenzo. Distributed training of graph convolutional networks. *IEEE Transactions on Signal and Information Processing over Networks*, 7: 87–100, 2020.
- Pranay Sharma, Rohan Panda, Gauri Joshi, and Pramod Varshney. Federated minimax optimization: Improved convergence analyses and algorithms. In *International Conference on Machine Learning*, pp. 19683–19730. PMLR, 2022.
- Kai Shu, Suhang Wang, and Huan Liu. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 312–320, 2019.
- Sebastian U Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2018.

- Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. FedNest: Federated bilevel, minimax, and compositional optimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 21146–21179. PMLR, 17–23 Jul 2022.
- Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the third ACM international workshop on edge systems, analytics and networking*, pp. 61–66, 2020.
- Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*.
- Xiyan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International conference on machine learning*, pp. 23341–23362. PMLR, 2022.
- Mark Weber, Jie Chen, Toyotaro Suzumura, Aldo Pareja, Tengfei Ma, Hiroki Kanezashi, Tim Kaler, Charles E Leiserson, and Tao B Schardl. Scalable graph learning for anti-money laundering: A first look. *arXiv preprint arXiv:1812.00076*, 2018.
- Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*, 2019.
- Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.
- Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahon, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pp. 10334–10343. PMLR, 2020a.
- Blake E Woodworth, Kumar Kshitij Patel, and Nati Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33:6281–6292, 2020b.
- 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.
- 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.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
- Pengwei Xing, Songtao Lu, Lingfei Wu, and Han Yu. Big-fed: Bilevel optimization enhanced graph-aided federated learning. *IEEE Transactions on Big Data*, pp. 1–12, 2022. doi: 10.1109/TBDATA.2022.3191439.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. *Advances in Neural Information Processing Systems*, 26, 2013.
- Yuhang Yao, Weizhao Jin, Srivatsan Ravi, and Carlee Joe-Wong. Fedgcn: Convergence-communication tradeoffs in federated training of graph convolutional networks. *Advances in neural information processing systems*, 36:79748–79760, 2023.
- Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10737–10745, 2021.

- Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019a.
- Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5693–5700, 2019b.
- Zhuoning Yuan, Zhishuai Guo, Yi Xu, Yiming Ying, and Tianbao Yang. Federated deep auc maximization for heterogeneous data with a constant communication complexity. In *International Conference on Machine Learning*, pp. 12219–12229. PMLR, 2021.
- Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *CoRR*, abs/2010.08982, 2020. URL <https://arxiv.org/abs/2010.08982>.
- Yi Zhang, Yuying Zhao, Zhaoqing Li, Xueqi Cheng, Yu Wang, Olivera Kotevska, Philip S Yu, and Tyler Derr. A survey on privacy in graph neural networks: Attacks, preservation, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- Zaixi Zhang, Qi Liu, Zhenya Huang, Hao Wang, Chee-Kong Lee, and Enhong Chen. Model inversion attacks against graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8729–8741, 2022.
- Chuanpan Zheng, Xiaoliang Fan, Shirui Pan, Haibing Jin, Zhaopeng Peng, Zonghan Wu, Cheng Wang, and Philip S Yu. Spatio-temporal joint graph convolutional networks for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(1):372–385, 2023.

A GRAPH NEURAL NETWORK

$$\begin{aligned}
\text{GCN: } \mathbf{h}_v^{(l)} &= \phi \left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{1}{\sqrt{\tilde{d}_v \tilde{d}_u}} \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)} \right), \quad \tilde{A} = A + I, \quad \tilde{d}_v = \sum_u \tilde{A}_{vu}. \\
\text{GIN: } \mathbf{h}_v^{(l)} &= \phi \left(\mathbf{W}^{(l)} \left((1 + \varepsilon) \mathbf{h}_v^{(l-1)} + \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(l-1)} \right) \right), \quad \varepsilon^{(l)} \text{ is fixed or learnable.} \\
\text{GraphSAGE-mean: } \mathbf{h}_v^{(l)} &= \phi \left(\mathbf{W}^{(l)} \cdot \frac{1}{|\mathcal{N}(v) \cup \{v\}|} \sum_{u \in \mathcal{N}(v) \cup \{v\}} \mathbf{h}_u^{(l-1)} \right).
\end{aligned} \tag{20}$$

We denote $h_v^{(l)} = \phi(\tilde{h}_v^{(l)}) = \phi(\mathbf{W}^{(l)} \hat{h}_v^{(l)})$.

Recent advances in GNN research have explored how aggregation functions (Xu et al., 2018) and spectral perspectives (Wang & Zhang, 2022) influence expressivity, enabling models to better capture higher-order dependencies and subtle structural cues. To enhance expressive power, various techniques have been proposed, including reverse message passing, port numbering, and ego IDs (Jaume et al., 2019; Sato et al., 2019; You et al., 2021). Financial crime detection provides a concrete example where such expressivity is indispensable, as transaction networks naturally form directed multigraphs with multiple transactions between the same entities (Cardoso et al., 2022; Kanezashi et al., 2022; Weber et al., 2018; 2019; Nicholls et al., 2021). Building on these adaptations, Egressy et al. (2024) demonstrates both theoretically and empirically that expressive GNN architectures substantially improve the detection of money laundering patterns in transaction graphs. However, in realistic scenarios where data is privacy-sensitive, training such expressive GNNs in a federated setting remains a critical and largely unexplored challenge, particularly in privacy-sensitive domains such as cross-institution financial networks.

B ANALYSIS

We first establish the assumptions we need and then present analysis in this section. A function f is said to be C_0 -Lipschitz continuous if for all $\mathbf{x}, \mathbf{x}'$ in its domain, $\|f(\mathbf{x}) - f(\mathbf{x}')\| \leq C_0 \|\mathbf{x} - \mathbf{x}'\|$. A differentiable function f is C_1 -smooth if its gradient is Lipschitz continuous, meaning $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \leq C_1 \|\mathbf{x} - \mathbf{x}'\|$ for all $\mathbf{x}, \mathbf{x}'$. Let superscript H denote computing forward and backward propagation using the embedding estimator H , while superscript h denotes replacing H estimator with true embeddings of sampled data. We make the following assumption throughout this paper.

Assumption B.1.

- i) f, g, h are C_0 -Lipschitz and C_1 -smooth over $\mathbf{W}$, and $\phi(\cdot)$ is also C_0 -Lipschitz and C_1 -smooth.
- ii) Over a batch B , $\mathbb{E} \left\| \hat{\nabla} F_i^h(\mathbf{w}; B) - \hat{\nabla} F_i(\mathbf{w}) \right\|^2 \leq \sigma^2$, and also $\|\hat{\nabla} F_i^H(\mathbf{w}; B)\| \leq D^2$.
- iii) There exists $\mathbf{W}_*$ such that $F(\mathbf{W}) \geq F(\mathbf{W}_*) > -\infty, \forall \mathbf{W}$.
- iii) $\|\mathbf{W}\|^2 \leq C_W^2, \|H(\cdot)\|^2 \leq C_H^2$,

B.1 PROOF OF LEMMA 3.1

Let $p_i(v)$ denote the probability for a node v to be sampled in client i , then we know that $p_i(v) > B_0/|\mathcal{E}_i|$ for edge based tasks, where B_0 is the number of seed edges, while $\mathcal{E}_i$ is the total number of edges on client i . Let $\tilde{h}_{r,k}^{(l)}(v)$ be the l -th layer's deterministic pre-activation embedding computed using model $\mathbf{W}_{r,k} := \frac{1}{N} \sum_i \mathbf{W}_{r,i,k}$ and all neighbors. Let $c(v)$ denote the hosting client of node v .

For the l -th layer with $l > 1$, we have

$$\begin{aligned}
& \mathbb{E} \|\tilde{H}_{r,i,k}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v)\|^2 \\
&= p(v) \mathbb{E} \left\| (1 - \gamma) \tilde{H}_{r,i,k-1}^{(l)}(v) + \gamma \frac{1}{n_{r,i,k}(v)} \left(\mathbf{w}_{r,i,k-1}^{(l)} \cdot \sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k-1}^{(l-1)}(u) \right) \right. \\
&\quad \left. - \tilde{h}_{r,k-1}^{(l)}(v) + \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v)) \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k}^{(l)}(v) \right\|^2 \\
&= p(v) \mathbb{E} \left\| (1 - \beta) (\tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{r,k-1}^{(l)}(v)) \right. \\
&\quad \left. + \gamma \left(\mathbf{w}_{r,i,k-1}^{(l)} \cdot \frac{1}{n_{r,i,k}} \left(\sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k-1}^{(l-1)}(u) \right) - \tilde{h}_{r,k-1}^{(l)}(v) \right) \right. \\
&\quad \left. + \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v)) \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k}^{(l)}(v) \right\|^2 \\
&\leq (1 + \frac{\gamma}{4}) p(v) \mathbb{E} \left\| (1 - \beta) (\tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{r,k-1}^{(l)}(v)) \right. \\
&\quad \left. + \gamma \left(\mathbf{w}_{r,i,k-1}^{(l)} \cdot \frac{1}{n_{r,i,k}(v)} \left(\sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k-1}^{(l-1)}(u) \right) - \tilde{h}_{r,k-1}^{(l)}(v) \right) \right\|^2 \\
&\quad + (2 + \frac{4}{\gamma} + \frac{4}{\gamma p(v)}) p(v) \left\| \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v)) (1 + \frac{\gamma p(v)}{4}) \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2
\end{aligned} \tag{21}$$

which then leads to

$$\begin{aligned}
& \mathbb{E} \|\tilde{H}_{r,i,k}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v)\|^2 \leq (1 + \frac{\gamma}{4}) p(v) \mathbb{E} \left\| (1 - \gamma) (\tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{r,k-1}^{(l)}(v)) \right. \\
&\quad \left. + \gamma \left(\mathbf{w}_{r,i,k-1}^{(l)} \cdot \left(\frac{1}{n_{r,i,k}(v)} \sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k-1}^{(l-1)}(u) - \frac{1}{n_i(v)} \sum_{u \in \mathcal{N}_i(v)} H_{r,i,k-1}^{(l-1)}(u) \right) \right) \right\|^2 \\
&\quad + \gamma \left(\mathbf{w}_{r,i,k-1}^{(l)} \cdot \left(\frac{1}{n_i(v)} \sum_{u \in \mathcal{N}_i(v)} H_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l)}(v) \right) \right\|^2 \\
&\quad + \frac{5}{\gamma} \left\| \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v)) (1 + \frac{\gamma p(v)}{4}) \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2.
\end{aligned} \tag{22}$$

Then noting

$$\mathbb{E}_{r,k-1} \left(\frac{1}{n_{r,i,k}(v)} \sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k-1}^{(l-1)}(u) - \frac{1}{n_i(v)} \sum_{u \in \mathcal{N}_i(v)} H_{r,i,k-1}^{(l-1)}(u) \right) = 0, \tag{23}$$

we have

$$\begin{aligned}
& \mathbb{E} \|\tilde{H}_{r,i,k}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v)\|^2 \\
& \leq (1 + \frac{\gamma}{4})p(v)\mathbb{E} \left\| (1 - \gamma)(\tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{r,k-1}^{(l)}(v)) \right. \\
& \quad \left. + \gamma \left(\mathbf{W}_{r,i,k-1}^{(l)} \cdot \frac{1}{n_i(v)} \sum_{u \in \mathcal{N}(v)} H_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l)}(v) \right) \right\|^2 \\
& \quad + \gamma^2 G^2 + \frac{5}{\gamma} \left\| \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v))(1 + \frac{\gamma p(v)}{4})\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 \\
& \leq (1 + \frac{\gamma}{4})^2(1 - \gamma)^2 p(v)\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(u) - \tilde{h}_{i,k-1}^{(l)}(u) \right\|^2 \\
& \quad + (1 + \frac{4}{\gamma})\gamma^2 \left\| \left(\mathbf{W}_{r,i,k-1}^{(l)} \cdot \frac{1}{n_i(v)} \sum_{u \in \mathcal{N}(v)} H_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l)}(v) \right) \right\|^2 + \gamma^2 G^2 \\
& \quad + \frac{5}{\gamma} \left\| \tilde{h}_{r,k-1}^{(l)}(v) - \tilde{h}_{r,k}^{(l)}(v) \right\|^2 + (1 - p(v))(1 + \frac{\gamma p(v)}{4})\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 \\
& \leq (1 - \frac{\gamma p(v)}{4})\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(u) - \tilde{h}_{i,k-1}^{(l)}(u) \right\|^2 + 8\gamma C_W^2 \frac{1}{n_i(v)} \sum_{u \in \mathcal{N}(v)} \|\tilde{H}_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l-1)}(u)\|^2 \\
& \quad + \gamma^2 C_W^2 C_H^2 + \frac{8}{\gamma} C^2 \|\mathbf{W}_{r,k} - \mathbf{W}_{r,k-1}\|^2.
\end{aligned} \tag{24}$$

where C is a Lipschitz constant of $\tilde{h}_{r,k}^{(l)}$ over $\mathbf{W}$, depending on the constants in Assumption B.1. Rearranging the terms, we have

$$\begin{aligned}
& \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 \leq \frac{4 \left(\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 - \mathbb{E} \left\| \tilde{H}_{r,i,k}^{(l)}(v) - \tilde{h}_{i,k}^{(l)}(v) \right\|^2 \right)}{\gamma p(v)} \\
& \quad + \frac{32C_W^2}{p(v)n_i(v)} \sum_{u \in \mathcal{N}(v)} \|\tilde{H}_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l-1)}(u)\|^2 + \frac{4}{p(v)}\gamma C_W^2 C_H^2 + \frac{32}{\gamma^2 p(v)} C^2 \|\mathbf{W}_{r,k} - \mathbf{W}_{r,k-1}\|^2.
\end{aligned} \tag{25}$$

Note that $\tilde{H}_{r,i,0}^{(l)} = \tilde{H}_{r-1,i,K}^{(l)}$, and $\tilde{h}_{r,0}^{(l)} = \tilde{h}_{r-1,K}^{(l)}$. Let $p := \min_{u \in \mathcal{V}} p(u)$. Taking the telescoping sum, we have

$$\begin{aligned}
& \frac{1}{RK} \sum_r \sum_k \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 \leq O \left(\frac{4(L+1)C^2}{\gamma p RK} \right. \\
& \quad \left. + \frac{4}{p} \gamma (L+1)G^2 + \frac{32}{\gamma^2 p} (L+1)C^2 \|\mathbf{W}_{r,k} - \mathbf{W}_{r,k-1}\|^2 \right).
\end{aligned} \tag{26}$$

And thus using Lipschitz of $\phi(\cdot)$, we obtain

$$\begin{aligned}
& \frac{1}{RK} \sum_r \sum_k \mathbb{E} \left\| H_{r,i,k-1}^{(l)}(v) - h_{i,k-1}^{(l)}(v) \right\|^2 \leq O \left(\frac{4(L+1)C^2}{\gamma p RK} \right. \\
& \quad \left. + \frac{4}{p} \gamma (L+1)G^2 + \frac{32}{\gamma^2 p} (L+1)C^2 \|\mathbf{W}_{r,k} - \mathbf{W}_{r,k-1}\|^2 \right).
\end{aligned} \tag{27}$$

B.2 PROOF OF LEMMA 3.2

Let superscript H denote computing forward and backward propagation using the embedding estimator H , while superscript h denotes replacing H estimator with true embeddings of sampled data.

By the update rule of the gradient estimator, we have that

$$\begin{aligned}
& \|\bar{G}_{r,k} - \nabla F(\mathbf{W}_{r,k})\|^2 \\
& \leq (1 + \frac{\beta}{4}) \left\| (1 - \beta)\bar{G}_{r,k-1} + \beta \frac{1}{N} \sum_{i=1}^N \left(\hat{\nabla} F_i(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) \right) - \nabla F(\mathbf{W}_{r,k-1}) \right\|^2 \\
& \quad + (1 + \frac{4}{\beta}) C^2 \|\mathbf{W}_{r,k-1} - \mathbf{W}_{r,k}\|^2 \\
& = (1 + \frac{\beta}{4}) \left\| (1 - \beta)(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1})) \right. \\
& \quad \left. + \beta \frac{1}{N} \sum_{i=1}^N \left(\hat{\nabla} F_i^H(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) \right) \right. \\
& \quad \left. + \beta \frac{1}{N} \sum_{i=1}^N \left(\hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \nabla F(\mathbf{W}_{r,k-1}) \right) \right\|^2 + (1 + \frac{4}{\beta}) C^2 \|\mathbf{W}_{r,k-1} - \mathbf{W}_{r,k}\|^2 \\
& = (1 + \frac{\beta}{4}) \left\| (1 - \beta)(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1})) \right. \\
& \quad \left. + \beta \frac{1}{N} \sum_{i=1}^N \left(\hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \nabla F(\mathbf{W}_{r,k-1}) \right) \right\|^2 \\
& \quad + (1 + \frac{4}{\beta}) \beta^2 \frac{1}{N} \sum_{i=1}^N \left\| \left(\hat{\nabla} F_i^H(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) \right) \right\|^2 \\
& \quad + (1 + \frac{4}{\beta}) C^2 \|\mathbf{W}_{r,k-1} - \mathbf{W}_{r,k}\|^2 \\
& \leq (1 + \frac{\beta}{4})^2 \left\| (1 - \beta)(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1})) \right. \\
& \quad \left. + \beta \frac{1}{N} \sum_{i=1}^N \left(\hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \nabla F(\mathbf{W}_{r,k-1}) \right) \right\|^2 + (1 + \frac{4}{\beta}) \beta^2 \frac{1}{N} \sum_{i=1}^N \|\mathbf{W}_{r,k-1} - \mathbf{W}_{r,i,k-1}\|^2 \\
& \quad + (1 + \frac{4}{\beta}) \beta^2 \frac{1}{N} \sum_{i=1}^N \left\| \left(\hat{\nabla} F_i^H(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) \right) \right\|^2 \\
& \quad + (1 + \frac{4}{\beta}) C^2 \|\mathbf{W}_{r,k-1} - \mathbf{W}_{r,k}\|^2 \\
& \leq (1 - \frac{\beta}{4}) \|\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1})\|^2 + \frac{\beta^2 \sigma^2}{N} + C^2 5\beta \eta^2 K^2 D^2 \\
& \quad + 5\beta \frac{1}{N} \sum_{i=1}^N \|\hat{\nabla} F_i^{H'}(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k})\|^2 + \beta^3 K^2 C_H^2 \\
& \quad + \frac{5C^2 \eta^2}{\beta} \|\bar{G}_{r,k-1}\|^2,
\end{aligned} \tag{28}$$

where superscript H' denoting plugging all H estimators at iteration (r, i, k) . For remote nodes, the difference between current estimator and previous estimator is bounded by $\beta^2 K^2 C_H^2$. And

$$\|\hat{\nabla} F_i^H(\mathbf{W}_{r,i,k-1}; B_{r,i,k}) - \hat{\nabla} F_i^h(\mathbf{W}_{r,i,k-1}; B_{r,i,k})\|^2 \leq \sum_{u \in B_{r,i,k}} C^2 m(u, l) \|\tilde{H}^{(l)r,i,k}(u) - \tilde{h}_{r,i,k}^{(l)}(u)\|^2, \tag{29}$$

with $m(u, l)$ is the weight for node u at layer l satisfying $\sum_{u \in B_{r,i,k}} m(u, l) = 1$.

Rearranging terms, applying telescoping sum and plugging Lemma 3.5,

$$\begin{aligned} \frac{1}{RK} \sum_r \sum_k \|(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1}))\|^2 &\leq O\left(\frac{1}{\beta RK} + \frac{1}{\gamma p RK}\right. \\ &\quad \left. + \frac{\beta}{N} + \frac{\gamma}{p} + \beta^2 K^2 + \gamma^2 K^2 + \frac{8C^2\eta^2}{\beta^2} \|\bar{G}_{r,k-1}\|^2\right). \end{aligned} \quad (30)$$

With $\eta = O(\beta)$, we have

$$\begin{aligned} \frac{1}{RK} \sum_r \sum_k \|(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1}))\|^2 &\leq O\left(\frac{1}{\beta RK} + \frac{1}{\gamma p RK}\right. \\ &\quad \left. + \frac{\beta}{N} + \frac{\gamma}{p} + \beta^2 K^2 + \frac{\gamma^2 K^2}{p} + \frac{\eta^2}{\beta^2} \|\nabla F(\mathbf{W}_{r,k-1})\|^2\right). \end{aligned} \quad (31)$$

B.3 PROOF OF THEOREM 3.3

Using C_1 -smooth of F , we have

$$\begin{aligned} F(\mathbf{W}_r) &\leq F(\mathbf{W}_{r-1}) + \langle \nabla F(\mathbf{W}_{r-1}), \mathbf{W}_r - \mathbf{W}_{r-1} \rangle + \frac{C_1}{2} \|\mathbf{W}_r - \mathbf{W}_{r-1}\|^2 \\ &= F(\mathbf{W}_{r-1}) - \langle \nabla F(\mathbf{W}_{r-1}), \eta \frac{1}{N} \sum_i \sum_k G_{r,i,k}(\mathbf{W}_{r,i,k}) \rangle + \frac{C_1}{2} \|\mathbf{W}_r - \mathbf{W}_{r-1}\|^2 \\ &= F(\mathbf{W}_{r-1}) - \langle \nabla F(\mathbf{W}_{r-1}), \eta \frac{1}{N} \sum_i \sum_k \hat{\nabla} F_i(\mathbf{W}_{r-1}) \rangle + \frac{C_1}{2} \|\mathbf{W}_r - \mathbf{W}_{r-1}\|^2 \\ &\quad - \langle \nabla F(\mathbf{W}_{r-1}), \eta \frac{1}{N} \sum_i \sum_k G_{r,i,k} - \eta \frac{1}{N} \sum_i \sum_k \hat{\nabla} F_i(\mathbf{W}_{r-1}) \rangle \\ &= F(\mathbf{W}_{r-1}) - \eta K \|F(\mathbf{W}_{r-1})\|^2 + \frac{\eta}{2} K \|\nabla F(\mathbf{W}_{r-1})\|^2 \\ &\quad + \frac{\eta}{K} \left\| \frac{1}{N} \sum_i \sum_k G_{r,i,k} - \frac{1}{N} \sum_i \sum_k \hat{\nabla} F_i(\mathbf{W}_{r-1}) \right\|^2 \\ &\quad + \frac{\eta K}{NK} \sum_i \sum_k \|\mathbf{W}_{r-1} - \mathbf{W}_{r,i,k}\|^2 + \frac{C_1}{2} \eta^2 \left\| \frac{1}{N} \sum_i \sum_k G_{r,i,k} \right\|^2. \end{aligned} \quad (32)$$

Thus, plugging Lemma 3.2, we have

$$\begin{aligned} &\frac{1}{R} \sum_r \mathbb{E} \|\nabla F(\mathbf{W}_{r-1})\|^2 \\ &\leq O\left(\frac{F(\mathbf{W}_0) - F(\mathbf{W}_*)}{\eta RK} + \frac{1}{RK} \sum_r \sum_k \|(\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1}))\|^2 + \eta^2 K^2 D^2\right) \\ &\leq O\left(\frac{1}{\eta RK} + \frac{1}{\beta RK} + \frac{1}{\gamma p RK} + \frac{\beta}{N} + \frac{\gamma}{p} + \beta^2 K^2\right). \end{aligned} \quad (33)$$

C ANALYSIS UNDER DIFFERENTIAL PRIVACY

The analysis of Lemma 3.4 is based on Lemma 3.1. Within a round r , we have

$$\begin{aligned} \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 &\leq \frac{4 \left(\mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 - \mathbb{E} \left\| \tilde{H}_{r,i,k}^{(l)}(v) - \tilde{h}_{i,k}^{(l)}(v) \right\|^2 \right)}{\beta p(v)} \\ &+ \frac{32C_W^2}{p(v)n_i(v)} \sum_{u \in \mathcal{N}(v)} \left\| \tilde{H}_{r,i,k-1}^{(l-1)}(u) - \tilde{h}_{r,k-1}^{(l-1)}(u) \right\|^2 + \frac{4}{p(v)} \beta C_W^2 C_H^2 + \frac{32}{\beta^2 p(v)} C^2 \left\| \mathbf{W}_{r,k}^{(l)} - \mathbf{W}_{r,k-1}^{(l)} \right\|^2. \end{aligned} \quad (34)$$

Due to the noise to $\mathbf{W}$ at the global communication round, we have

$$\begin{aligned} &\mathbb{E} \left\| \tilde{H}_{r,i,K}^{(l)}(v) - \tilde{h}_{r,i,K}^{(l)}(v) \right\|^2 \\ &= \mathbb{E} \left\| \tilde{H}_{r,i,K}^{(l)}(v) - \tilde{h}_{r+1,i,0}^{(l)}(v) + \tilde{h}_{r+1,i,0}^{(l)}(v) - \tilde{h}_{i,K}^{(l)}(v) \right\|^2 \\ &\leq \mathbb{E} \left\| \tilde{H}_{r,i,K}^{(l)}(v) - \tilde{h}_{r+1,i,0}^{(l)}(v) \right\|^2 + C_H \sigma_1 + \sigma_1^2. \end{aligned} \quad (35)$$

As a result, taking the telescoping sum yields

$$\begin{aligned} &\frac{1}{RK} \sum_r \sum_k \mathbb{E} \left\| \tilde{H}_{r,i,k-1}^{(l)}(v) - \tilde{h}_{i,k-1}^{(l)}(v) \right\|^2 \\ &\leq O\left(\frac{1}{\gamma RK} + \gamma + \gamma^2 K^2 + \frac{\left\| \mathbf{W}_{r,k} - \mathbf{W}_{r,k-1} \right\|^2}{\gamma^2} + \frac{\sigma_1 + \sigma_1^2}{\gamma}\right), \end{aligned} \quad (36)$$

which concludes Lemma 3.4.

Similarly, based on the analysis of Lemma 3.2, we can have Lemma 3.5. Besides the noise in $\mathbf{W}$ as we handled above, it also depends on noise to H , which would add a σ_0^2 to (29). Also, gradient estimator G is added a noise of $\mathcal{G}(0, \sigma_2^2)$. Therefore,

$$\begin{aligned} &\frac{1}{RK} \sum_r \sum_k \left\| (\bar{G}_{r,k-1} - \nabla F(\mathbf{W}_{r,k-1})) \right\|^2 \leq \\ &O\left(\frac{1}{\beta RK} + \frac{1}{\gamma RK} + \beta + \gamma + \beta^2 K^2 + \frac{\eta^2}{\beta^2} \left\| \nabla F(\mathbf{W}_{r,k-1}) \right\|^2 + \sigma_0^2 + \sigma_1^2 + \sigma_1/\beta + \sigma_2^2 + \sigma_2/\beta\right) \end{aligned} \quad (37)$$

Then Theorem 3.6 follows as

$$\frac{1}{R} \sum_r \mathbb{E} \left\| \nabla F(\mathbf{W}_{r-1}) \right\|^2 \leq O\left(\frac{1}{\eta RK} + \beta + \beta^2 K^2 + \sigma_0^2 + \sigma_1^2 + \sigma_1/\beta + \sigma_2^2 + \sigma_2/\beta\right). \quad (38)$$

Noise Level in Differential Privacy We focus on the differential privacy guarantee of shared embeddings. Following the common practice of clipping McMahan et al. (2018), we clip the Euclidean norm each shared embedding to be less than 10 by and only share node embeddings when there are at least 20 neighbor data points have been involved to update its embedding. Then we set δ in (ϵ, δ) to be 0.01, thus the corresponding upper bound of ϵ regarding to σ_0 is as follows in the 20 communication rounds (each round has 1024 iterations): $\sigma_0 = 1e - 2 : \epsilon \leq 50004$, $\sigma_0 = 0.1 : \epsilon \leq 505$, $\sigma_0 = 0.3 : \epsilon \leq 61$, $\sigma_0 = 0.5 : \epsilon \leq 25$, $\sigma_0 = 0.7 : \epsilon \leq 15$, $\sigma_0 = 0.9 : \epsilon \leq 11$, $\sigma_0 = 2 : \epsilon \leq 5$.

D DATA STATISTICS

Data statistics is summarized in Table 3. L/S denotes the ratio of the number of data on the largest client over that on the smallest client.

For the two small datasets, training days are 0-7, validation day is 8, and testing days are 9-13. For the two medium datasets, training days are 0-13, validation day is 14, and testing days are 15-26. For the two large datasets, training days are 0-94, validation day is 95, and testing days are 96-162.

Table 3: Statistics of the Datasets

	# of Edges	# of Nodes	# of Clients	L/S
HI-Small	5,078,345	515,088	4	1.37
HI-Medium	31,898,238	2,077,023	6	1.52
HI-Large	179,702,229	2,116,168	32	3.47
LI-Small	6,924,049	705,907	4	1.28
LI-Medium	31,251,483	2,032,095	6	1.55
LI-Large	179,702,229	2,070,980	16	3.15

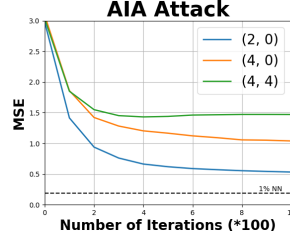


Figure 4: AIA attack under different hop-1 and hop-2 size. Legends are in the form of (hop-1, hop-2).

E EXPERIMENTS ON ATTRIBUTE INFERENCE ATTACK

In this subsection, we focus on the HI-Small dataset to study attribute inference attacks (AIA) on our methods. We concentrate on AIA because it represents a realistic privacy threat in federated GNNs. Unlike MIA or GSR Zhang et al. (2024), AIA better models adversaries in financial networks who may access partial embeddings and aim to recover sensitive node attributes. We assume the attacker knows a node’s membership in the targeted bank, along with a subset of nodes and connections in other banks. Leveraging this information together with shared embeddings, the attacker attempts to reconstruct the remaining unknown features. Reconstruction quality is evaluated using the following loss function.

$$\arg \min_{x'} \|h^{(L)}(x'; B \cup \{x\}) - h^{(L)}(x; B \cup \{x\})\|^2. \quad (39)$$

We then measure the reconstruction quality using the Mean Squared Error (MSE) between the true features x and the reconstructed features x' .

The results, shown in Figure 4, indicate that embeddings computed with more neighbors are harder to reconstruct. Additionally, we benchmarked the average distance to the top 1% nearest neighbors in the dataset. The attribute inference attack (AIA) fails to reconstruct raw features within this 1% threshold, verifying that sharing aggregated embeddings provides strong robustness against privacy attacks.

F USING EXTERNAL NEIGHBORS FOR NODE EMBEDDING AGGREGATION

Let the subscript r, i, k denote round r at machine i and iteration k . The forward pass for a node using a moving average is defined as:

$$\begin{aligned} \tilde{H}_{r,i,k}^{(l)}(v) &= (1 - \gamma)\tilde{H}_{r,i,k-1}^{(l)}(v) + \gamma \frac{1}{n_{r,i,k}(v)} \mathbf{W}_{r,i,k-1}^{(l,n)} \cdot \left(\sum_{u \in \mathcal{N}_{r,i,k}(v)} H_{r,i,k}^{(l-1)}(u) + \sum_{u \in \mathcal{N}'_{r,i,K}(v,l)} H_{r-1,c(u),K}^{(l-1)}(u) \right), \\ H_{r,i,k}^{(l)}(v) &= \phi(\tilde{H}_{r,i,k}^{(l)}(v)), \end{aligned} \quad (40)$$

Here, $\mathcal{N}_{r,i,k}(v)$ denotes the local neighbors in the current batch, while $\mathcal{N}'_{r,i,K}(v,l)$ corresponds to neighbors from other clients whose embeddings were previously shared. Note that $\mathcal{N}'_{r,i,K}(v,l)$

depends on l since that we define a node's neighbors varying with respect to layers to utilize only high level embeddings from remote client to save communication time and reduce privack risk. The embeddings $H^{(l-1)}r - 1, c(u), K(u)$ are retrieved from other clients since u is a remote node. And $n_{r,i,k}(v) = |\mathcal{N}_{r,i,k}(v) \cup \mathcal{N}^l_{r,i}, K(v, l)|$. Input embeddings are set as $H^{(0)}(\cdot) = h^{(0)}(\cdot)$.

In addition to sharing W, G, H as described in the main text, we also share portions of the chain rule with other clients to enable gradient computation. These shared components can similarly use a moving average estimator to control variance, though in practice sharing them is optional.