# CoCo: COllaborate via COarsening for Personalized Federated Graph Learning

**Anonymous authors**
Paper under double-blind review

## Abstract

Federated graph learning (FGL) enables collaborative training of graph neural networks without sharing raw data, but suffers from missing global structure and significant data heterogeneity. While personalized methods mitigate heterogeneity, existing approaches rely on noisy, training-dependent similarity signals and lack principled mechanisms to determine *who should collaborate with whom*. We propose CoCo, a personalized FGL framework that infers a client collaboration graph via privacy-preserving global structure reconstruction. By decoupling collaboration from training, CoCo enables a stable, one-shot estimation of collaboration patterns, avoiding the instability and communication overhead of iterative similarity learning. We provide theoretical guarantees on the stability of the inferred global structure under reconstruction error, the propagation of this stability to the induced collaboration graph, and a graph-smoothing property of the personalized collaboration step. From a privacy perspective, CoCo combines geometric non-identifiability with differential privacy to limit information leakage. Experiments on diverse benchmarks demonstrate consistent improvements over strong federated and personalized baselines, along with robustness to heterogeneity, noise, and privacy-utility trade-offs.

## 1 Introduction

In many high-stakes domains, graph data are siloed across organizations, while privacy and regulatory constraints prohibit centralization. Federated graph learning (FGL) Xie et al. (2021); Zhang et al. (2021); Li et al. (2020) enables multiple clients to collaboratively train graph-based models while keeping their data localized. This setting naturally arises in domains such as social networks, recommendation systems, and biomedical applications, where graph-structured data is distributed across entities and cannot be centralized due to privacy, regulatory, or ownership constraints. However, restricting direct data access introduces two core challenges: (**P1**) the absence of a global graph structure across clients, which limits the ability to exploit cross-client dependencies, and (**P2**) data heterogeneity, where client distributions differ significantly, degrading the performance of a single global model. Personalized Federated Learning (pFL) Ghosh et al. (2020); Mansour et al. (2020); Kotelevskii et al. (2022); Li et al. (2022); Deng et al. (2021); Tan et al. (2022); Allouah et al. (2024) addresses heterogeneity by learning client-specific models, but a central question remains unresolved: *which clients should collaborate, and to what extent?* Existing methods attempt to answer this by estimating similarity between clients using model updates, gradients, or auxiliary representations. However, these approaches suffer from deeper limitations: (**P3**) they rely on implicit and training-dependent similarity signals derived from model updates, which are inherently noisy, sensitive to early training dynamics, and require repeated communication, leading to unstable and inefficient collaboration patterns, and (**P4**) many collaboration-based methods do not explicitly characterize the information leakage associated with the signals used to infer client relationships. In this work, we propose CoCo, a personalized federated graph learning framework that addresses these challenges through a three-step pipeline (Figure 1): (**Step 1**) reconstruct a global graph structure from privacy-preserving anchor-based signals to address (P1), (**Step 2**) coarsen this structure to infer a client-level collaboration graph that captures meaningful relationships between clients, addressing (P3), and (**Step 3**) perform graph-guided personalized aggregation to adaptively balance collaboration and local specialization, addressing (P2). By separating structure inference from model optimization, CoCo avoids the instability of training-dependent similarity estimation and enables a stable, one-shot construction of collaboration patterns.



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

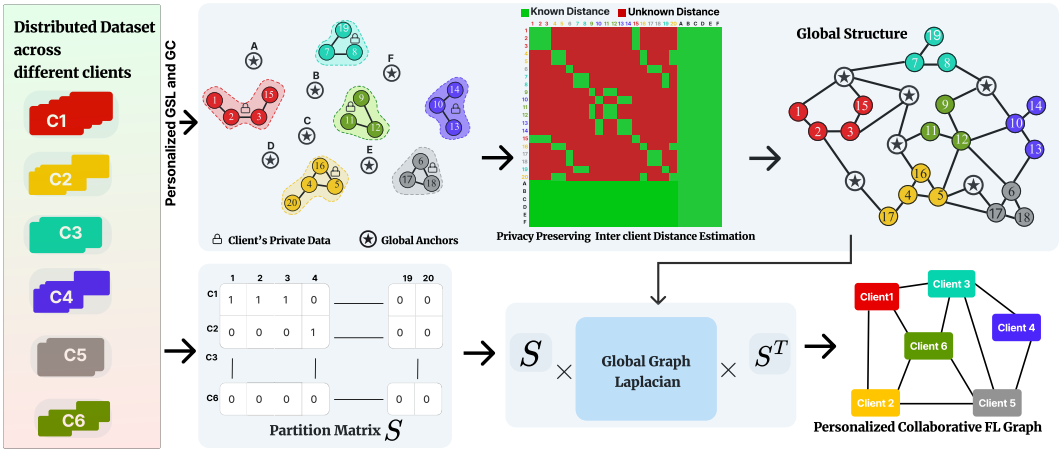



Figure 1: Pipeline of CoCo: Each client's data remain private, resulting in incomplete cross-client structural information. CoCo first recovers the global node-level structure $\widehat{L}_G$, and then uses the size-normalized client projection $P = (S^\top S)^{-1} S^\top$ to obtain the client-level coarsened operator $\widehat{L}_C = P \widehat{L}_G P^\top$. The resulting inter-client affinities define the collaboration graph used for personalized federated learning.

We further provide *theoretical guarantees* supporting this pipeline. Specifically, we show that the recovered global structure is stable under embedding reconstruction error, that this stability propagates to the client-level structure, and that the implemented collaboration step performs graph smoothing without increasing normalized graph variation. These results establish a principled foundation for structure-driven collaboration in federated settings. From a privacy perspective, CoCo addresses (P4) through two complementary mechanisms. First, under the conditions of Lemma 1 and Theorem 5, the anchor-distance representation is generically non-injective when the number of anchors is smaller than the feature dimension, preventing unique exact reconstruction from the released distances alone. Since geometric non-identifiability does not rule out approximate reconstruction or other inference attacks, we additionally apply node-level approximate local differential privacy to the released representation. Our main contributions are as follows:

1. We *redefine client collaboration* in federated graph learning as a coarsened projection of an underlying global geometry, shifting from training-dependent similarity estimation to a structure-driven paradigm.
2. We show that *collaboration can be constructed in a one-shot manner*, decoupled from model training, leading to stable and communication-efficient personalization.
3. We provide a *unified theoretical analysis* establishing stability of the recovered structure, its propagation to the client-level representation, and a graph-smoothing guarantee for the implemented collaboration step.
4. We introduce a *dual privacy mechanism* combining provable geometric non-identifiability against exact feature reconstruction with local differential privacy.
5. We demonstrate strong empirical performance across diverse datasets and tasks and further validate robustness under non-stationary clients, noisy clients, and privacy-utility trade-offs.

## 2 Background and Related Work

**Federated Learning and Personalized Federated Learning.** Federated learning (FL) enables multiple clients to train a shared model without exchanging raw data. FedAvg McMahan et al. (2017a) alternates local training with weighted server aggregation, while methods such as MOON Li et al. (2021) improve robustness to non-IID data through contrastive regularization. Nevertheless, a single global model can remain vulnerable to strong client heterogeneity.

Personalized federated learning (pFL) instead learns client-specific models while transferring knowledge across clients. Among general pFL methods, pFedGraph Ye et al. (2023) learns a client collaboration graph from training-dependent model similarities in standard sample-based FL, without modeling graph-structural collaboration; we therefore treat it as conceptually related rather than directly transferable to subgraph FGL. In contrast, CoCo constructs collaboration from cross-client data geometry, independently of model optimization, to recover missing structure in FGL. Graph-focused methods address personalization differently: FedSage+ Zhang et al. (2021) generates missing

cross-client neighbors; AdaFGL Li et al. (2024a) performs shared learning followed by client adaptation; FedIIH Yu et al. (2025) models inter- and intra-client heterogeneity; and FedGTA Li et al. (2024b) uses topology-aware statistics for personalized aggregation. Fed-PUB Baek et al. (2023) aggregates local GNNs using functional similarity, FedTAD Zhu et al. (2024) uses topology-aware data-free distillation, and FedTGP Zhang et al. (2024) learns adaptive global prototypes. These methods rely on training-dependent similarities, auxiliary statistics, generators, or latent representations, often requiring repeated communication. CoCo instead infers a one-shot collaboration structure while keeping raw client data local. Information-sharing patterns and privacy scope are detailed in Appendix E.

**Graphs and Graph Coarsening.** A graph is defined as $\mathcal{G} = (V, E, X)$, where $V = \{v_1, \ldots, v_N\}$ is the set of nodes, $E \subseteq V \times V$ is the set of edges, and $X \in \mathbb{R}^{N \times d}$ is the node feature matrix. Let $A \in \mathbb{R}^{N \times N}$ denote the corresponding adjacency matrix. The degree matrix is $D = \mathrm{Diag}(A\mathbf{1})$, with $D_{ii} = \sum_j A_{ij}$, and the unnormalized graph Laplacian is $L = D - A$.

Graph coarsening (GC) aims to reduce a graph $\mathcal{G}$ with $N$ nodes to a smaller graph $\mathcal{G}'$ with $N' \ll N$ supernodes Kataria et al. (2026; 2024). This reduction is represented by a coarsening matrix $\Pi \in \mathbb{R}^{N' \times N}$, where the nonzero entries of $\Pi$ encode the assignment or contribution of original nodes to supernodes. The resulting graph $\mathcal{G}'$ provides a compact representation of $\mathcal{G}$ while preserving relevant structural properties. Further details on graphs and graph coarsening are provided in Appendix A.1.

## 3 Methodology

We propose CoCo, a personalized federated graph learning framework that learns a client collaboration graph through a three-stage pipeline: (i) privacy-preserving global structure inference, (ii) graph coarsening to obtain a client-level collaboration graph, and (iii) graph-guided personalized aggregation. The pipeline and pseudocode are provided in Figure 1 and Algorithm 1, respectively.

*Goal:* Given $M$ clients $\{C_1, \ldots, C_M\}$, each client $C_m$ holds a local graph $\mathcal{G}_m = (V_m, E_m, X_m)$ with $X_m \in \mathbb{R}^{n_m \times d}$ and $n_m = |V_m|$. The total number of nodes is $N = \sum_{m=1}^{M} n_m$. The goal is to learn client-specific models while leveraging cross-client relationships without sharing raw data.

### 3.1 Privacy-Preserving Global Structure Inference (PP-GSI).

To address the absence of a global graph (P1) without transmitting raw client features, we infer a shared global structure from anchor-distance representations. These representations support cross-client structure inference while limiting direct information disclosure, their privacy properties are formalized in Section 4.

Unless otherwise stated, we first describe CoCo at the node level. For large graphs, we use the scalable instantiation in Sec. 3.4, where each client's local nodes are first coarsened into supernodes and the same PP-GSI and collaboration-graph construction are subsequently applied at the supernode level.

**Anchor-Based Representation.** Let $X_{\mathrm{priv}} \in \mathbb{R}^{N \times d}$ denote the matrix of private points distributed across the clients, and let $X_A \in \mathbb{R}^{K \times d}$ denote a set of $K$ shared reference anchors available to the server and all clients (details of the anchor generation process are outlined in Appendix A.2). For each private point $x_i$, the corresponding client computes the Euclidean distances to the public anchors, $d_{ij} = d(x_i, a_j) = \|x_i - a_j\|_2, \quad 1 \le i \le N, \quad 1 \le j \le K$, and transmits these private-to-anchor distances to the server. Distances among the public anchors are directly available because $X_A$ is public, whereas pairwise distances between inter-client private points are not observed. Using these partial distance observations, the server reconstructs the private point embeddings through anchored multidimensional scaling and subsequently estimates the global graph structure without requiring the transmission of raw client features. The missing geometric information is inferred from these partially observed distance blocks through a stress-minimization formulation based on classical multidimensional scaling (MDS) Nguyen and Shin (2016). Classical MDS embeds a set of points by minimizing stress over a fully observed distance matrix. Let $Z \in \mathbb{R}^{N \times d}$ denote a generic embedding, with $z_i$ denoting its $i$-th row and let $\Delta = [\delta_{ij}] \in \mathbb{R}^{N \times N}$ be the corresponding input distance matrix. The MDS stress is $\sigma(Z) = \sum_{i<j} \left(\|z_i - z_j\|_2 - \delta_{ij}\right)^2$, where $\delta_{ij}$ denotes the input Euclidean distance between points $i$ and $j$. SMACOF minimizes this objective through iterative majorization Malik et al. (2024); Koledoye et al. (2017); Zhou et al. (2021), using the update $Z^{(k)} = \mathcal{V}^{\dagger} B(Z^{(k-1)}) Z^{(k-1)}$, where $\mathcal{V}$ is the Laplacian associated with the SMACOF observation-

weight matrix $\Omega$, and $B(\cdot)$ is the corresponding majorization matrix. For the distributed anchored setting, define the combined matrix $X_{\text{aug}} = \left[X_{\text{priv}}^T X_A^T\right]^T \in \mathbb{R}^{(N+K)\times d}$. Following Malik et al. (2024); Di Franco et al. (2017), $\mathcal{V}$ and $B(X_{\text{aug}})$ are partitioned according to the private and anchor points as

$$\mathcal{V} = \begin{bmatrix} \mathcal{V}_{11} & \mathcal{V}_{12} \\ \mathcal{V}_{12}^\top & \mathcal{V}_{22} \end{bmatrix}, \qquad B(X_{\text{aug}}) = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^\top & B_{22} \end{bmatrix}, \tag{1}$$

where $\mathcal{V}_{11}, B_{11} \in \mathbb{R}^{N\times N}$, $\mathcal{V}_{12}, B_{12} \in \mathbb{R}^{N\times K}$, and $\mathcal{V}_{22}, B_{22} \in \mathbb{R}^{K\times K}$. The explicit form of $\mathcal{V}$ is

$$\mathcal{V} = \begin{bmatrix} KI_N & -\mathbf{1}_{N,K} \\ -\mathbf{1}_{N,K}^\top & (N+K)I_K - \mathbf{1}_{K,K} \end{bmatrix}.$$

Holding the public anchors fixed, the private point embeddings are reconstructed iteratively using the anchored-SMACOF update

$$\widehat{X}_{\text{priv}}^{(k)} = \frac{1}{K}\left(B_{11}\widehat{X}_{\text{priv}}^{(k-1)} + (B_{12} + \mathbf{1}_{N,K})X_A\right). \tag{2}$$

**Global Structure and Laplacian Recovery.** After obtaining the reconstructed embeddings $\widehat{X}_{\text{priv}}$ from Eq. 2, we infer the global graph over the $N$ private points. Following the graph-signal smoothness principle Kalofolias (2016), we solve

$$\widehat{A} = \arg\min_{A\in\mathcal{S}_A} \sum_{i,j=1}^{N} A_{ij}\|\widehat{x}_i - \widehat{x}_j\|_2^2 + f(A), \tag{3}$$

where $\mathcal{S}_A = \left\{A \in \mathbb{R}_+^{N\times N} \mid A = A^\top,\ \text{diag}(A) = 0,\ A\mathbf{1} \succ 0\right\}$, and $f(A) = -\alpha_g \sum_i \log((A\mathbf{1})_i) + \beta_g\|A\|_F^2, \quad \alpha_g, \beta_g > 0$. The recovered global Laplacian is

$$\widehat{L}_G = \text{Diag}(\widehat{A}\mathbf{1}) - \widehat{A}. \tag{4}$$

This yields a global structure inferred entirely from privacy-preserving anchor-distance representations, without access to raw client data. The stability of this reconstruction is analyzed in Theorem 1.

**Theorem 1** (Stability of Learned Global Structure). *Let $X_{\text{priv}}, \widehat{X}_{\text{priv}} \in \mathbb{R}^{N\times d}$ denote the original and reconstructed private-point embeddings, respectively, with $\|\widehat{X}_{\text{priv}} - X_{\text{priv}}\|_F \le \varepsilon_X$. Let $A^\star$ and $\widehat{A}$ denote the unique minimizers of Eq. 3 obtained using $X_{\text{priv}}$ and $\widehat{X}_{\text{priv}}$, respectively, and define $L_G^\star = \text{Diag}(A^\star\mathbf{1}) - A^\star$. Assume that the rows of the two embedding matrices satisfy $\|x_i\|_2 \le R_X, \quad \|\widehat{x}_i\|_2 \le R_X, \quad i = 1,\dots,N$, and that the graph regularizer $f$ is $\mu$-strongly convex over $\mathcal{S}_A$. Then $\|\widehat{L}_G - L_G^\star\|_F \le c_G\varepsilon_X$, where $c_G = \frac{8R_X\sqrt{N(N+1)}}{\mu}$. (The proof is provided in Appendix D.1).*

### 3.2 Collaboration Graph via Coarsening

Having recovered the global structure $\widehat{L}_G$ (Equation 4), we next coarsen the node-level structure to the client level to obtain interactions between clients. Let $V_m$ denote the nodes owned by client $C_m$, with $n_m = |V_m|$, and let $S \in \{0,1\}^{N\times M}$ be the binary client-membership matrix, where $S_{im} = 1$ iff $v_i \in V_m$. Since each node belongs to exactly one client, $S^\top S = \text{Diag}(n_1,\dots,n_M)$. To account for different client sizes, we define the normalized client projection $P = (S^\top S)^{-1}S^\top \in \mathbb{R}^{M\times N}$, such that $P_{mi} = 1/n_m$ if $v_i \in V_m$ and 0 otherwise. The size-normalized client-level coarsened Laplacian operator is then $\widehat{L}_C = P\widehat{L}_G P^\top \in \mathbb{R}^{M\times M}$. For $m \ne \ell$, $(\widehat{L}_C)_{m\ell} = -\frac{1}{n_m n_\ell}\sum_{u\in V_m}\sum_{v\in V_\ell}\widehat{A}_{uv}$, where $\widehat{A}$ is the adjacency associated with $\widehat{L}_G$. Thus, $-(\widehat{L}_C)_{m\ell}$ represents the mean node-level affinity between clients $C_m$ and $C_\ell$, providing a training-independent measure of inter-client relatedness. The normalization prevents larger clients from receiving disproportionately large interaction strengths solely due to their size. Although $\widehat{L}_C$ is symmetric and positive semidefinite, the size normalization means that it does not, in general, have zero row sums. We therefore use it as a size-normalized coarsened Laplacian operator from which the client affinity graph is constructed below.

*Interpretation as Collaboration Graph.* The client-level operator $\widehat{L}_C$ provides a training-independent, one-shot notion of collaboration from the recovered global structure (**P3**). We next analyze how reconstruction errors propagate to this client-level representation.

**Theorem 2** (Client-Coarsening Stability). *Let $\widehat{L}_G$ and $L_G^\star$ denote the recovered and true global operators, respectively, and let $\widehat{L}_C = P\widehat{L}_G P^\top$ and $L_C^\star = PL_G^\star P^\top$, where $P = (S^\top S)^{-1}S^\top$. If $n_{\min} = \min_{m=1,\dots,M} n_m$, then $\|\widehat{L}_C - L_C^\star\|_F \le \|P\|_2^2 \|\widehat{L}_G - L_G^\star\|_F = \frac{1}{n_{\min}}\|\widehat{L}_G - L_G^\star\|_F \le \frac{c_G}{n_{\min}}\varepsilon_X$, where the last inequality follows from Theorem 1. (The proof is provided in Appendix D.2).*

Thus, perturbations in the recovered global structure propagate in a controlled manner to the client-level coarsened representation, while the normalized projection prevents the stability bound from growing with the largest client size.

### 3.3 Graph-Guided Personalized Learning

To handle heterogeneous client distributions (**P2**), we leverage the inferred collaboration graph to guide personalized aggregation across clients.

*Graph-Guided Aggregation.* From $\widehat{L}_C$, we recover the inter-client affinity matrix as $\bar{A}_C = \mathrm{Diag}(\mathrm{diag}(\widehat{L}_C)) - \widehat{L}_C$ and $A_C = [(\bar{A}_C + \bar{A}_C^\top)/2]_+$, where $[\cdot]_+$ denotes element-wise clipping at zero. Let $D_C = \mathrm{Diag}(A_C\mathbf{1})$ and define the symmetric normalized collaboration matrix $W = D_C^{-1/2} A_C D_C^{-1/2}$. Thus, $W = W^\top$, although, unlike a row-stochastic aggregation matrix, its rows are not required to sum to one. In implementation, a small positive constant is added to the degrees when necessary for numerical stability. At each communication round, client $C_m$ first performs local training to obtain $\widetilde{\theta}_m^{(t+1)}$, followed by graph-guided aggregation $\theta_m^{(t+1)} = \alpha\widetilde{\theta}_m^{(t+1)} + (1-\alpha)\sum_{\ell\neq m} W_{m\ell}\widetilde{\theta}_\ell^{(t+1)}$ (Figure 2), where $\alpha \in [0,1]$. Stacking the locally updated client parameters row-wise, $\widetilde{\Theta}^{(t+1)} = [(\widetilde{\theta}_1^{(t+1)})^\top, \dots, (\widetilde{\theta}_M^{(t+1)})^\top]^\top \in \mathbb{R}^{M\times p}$, the collaboration update becomes $\Theta^{(t+1)} = [\alpha I_M + (1-\alpha)W]\,\widetilde{\Theta}^{(t+1)}$. The parameter $\alpha$ controls the trade-off between local specialization ($\alpha = 1$) and collaboration ($\alpha = 0$). Hence, each client selectively incorporates information from structurally related clients, adapting to client heterogeneity while retaining its locally trained model. The collaboration step admits an exact graph-smoothing interpretation. Define the normalized collaboration Laplacian $\mathcal{L}_C = I_M - W$ and the collaboration energy $\mathcal{E}_C(\Theta) = \frac{1}{2}\mathrm{tr}(\Theta^\top \mathcal{L}_C \Theta)$. Intuitively, $\mathcal{E}_C$ measures the degree-normalized variation of client models over the inferred collaboration graph: it is large when strongly connected clients have substantially different model parameters. Graph smoothing therefore corresponds to reducing such graph-inconsistent variation while preserving the structure encoded by the collaboration graph. Since $W = D_C^{-1/2} A_C D_C^{-1/2}$, $\mathcal{L}_C$ is symmetric positive semidefinite with eigenvalues in $[0,2]$. The following result shows the effect of the actual CoCo collaboration step.

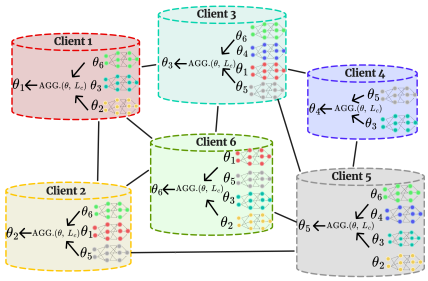


Figure 2: Personalized Model Update via Collaboration Graph.

**Theorem 3** (Graph-Smoothing Property of CoCo Collaboration). *Assume all clients have positive degree in $A_C$. For any locally updated parameters $\widetilde{\Theta} \in \mathbb{R}^{M\times p}$ and $\alpha \in [0,1]$, the CoCo aggregation $\Theta^+ = [\alpha I_M + (1-\alpha)W]\,\widetilde{\Theta}$ satisfies $\Theta^+ = \widetilde{\Theta} - (1-\alpha)\nabla\mathcal{E}_C(\widetilde{\Theta})$ and $\mathcal{E}_C(\Theta^+) \le \mathcal{E}_C(\widetilde{\Theta})$. Moreover, for $0 < \alpha < 1$, the inequality is strict whenever $\widetilde{\Theta}$ has a nonzero component outside $\ker(\mathcal{L}_C)$. (The proof is provided in Appendix D.3.)*

Thus, independently of the optimization procedure used during local GNN training, the subsequent CoCo collaboration stage is exactly a graph-smoothing step over the inferred client graph and cannot increase the normalized collaboration energy.

### 3.4 Scalability via Graph Coarsening

To scale CoCo to large graphs, we accelerate PP-GSI (Sec. 3.1) using graph coarsening, reducing the effective node set from $N$ to $N' \ll N$. The normalized client projection and subsequent collaboration-matrix construction remain identical to Section 3.2, with the node-level structure replaced by its supernode-level counterpart. Detailed implementation is provided in Appendix G. A detailed runtime analysis is provided in Appendix A.8.

**Remark 1** (Dynamic CoCo). *In settings where the underlying client data evolve over time, CoCo can update the collaboration graph by rerunning PP-GSI (Section 3.1) on the updated data and subsequently recomputing the client-level collaboration structure (Section 3.2).*

# 4 DUAL PRIVACY MECHANISM.

CoCo combines two complementary forms of privacy protection, (i) *geometric non-identifiability* of the anchor-distance representation in underdetermined regimes, and (ii) *local differential privacy* $(\epsilon_{\text{node}}, \delta_{\text{node}})$ applied to the complete anchor-distance vector released by each private node. Geometric non-identifiability limits unique exact reconstruction, whereas approximate LDP provides a formal stochastic indistinguishability guarantee for the complete released representation.

*Geometric Non-identifiability.* While Theorem 1 establishes the fidelity of the recovered structure, we next characterize the geometric non-identifiability of the released anchor-distance representation.

**Lemma 1** (Non-Identifiability of Anchor-Distance Representations, adapted from Malik et al. (2024))**.** *Let $X_{\text{priv}}$ denote the private data points and $X_A$ the set of $K$ anchors. If $K < d$, the original features $\{x_i\}$ cannot be uniquely reconstructed from the private point-anchor distances $\{\|x_i - a_j\|_2\}$. Hence, under these conditions, the anchor-distance representation does not uniquely determine the original private features. (See App. D.4 for proof.)*

CoCo's privacy model (Irrecoverability) is detailed in Appendix F, describing its adversary model, deterministic geometric non-identifiability guarantees (refer to Theorem 5), and the theoretical guarantees providing an information-theoretic leakage bound (refer to Lemma 2).

*Local Differential Privacy for Anchor-Distance Representations.* While geometric non-identifiability provides a deterministic guarantee against unique reconstruction in underdetermined regimes, it does not address all possible forms of information inference. For instance, as $K$ approaches or exceeds $d$, the deterministic ambiguity decreases and the released representation may become substantially more informative. Furthermore, when data exhibit additional structure, such as lying near a low-dimensional subspace Lichtenberg and Tasissa (2024); Liu and Yan (2011); Patel et al. (2015), the effective degrees of freedom may be significantly smaller than $d$, making the system more informative even when $K < d$. We therefore additionally apply approximate LDP to the complete private-to-anchor distance representation released by each private point. Our privacy unit is one private point $x_i$. Define its complete anchor-distance vector as, $\mathbf{d}_i = [d_{i1}, \ldots, d_{iK}]^\top \in \mathbb{R}_+^K$. Before transmission, the client clips the complete vector in $\ell_2$ norm: $\bar{\mathbf{d}}_i = \mathbf{d}_i \min\left\{1, \frac{C_{\text{vec}}}{\|\mathbf{d}_i\|_2}\right\}$, where $C_{\text{vec}} > 0$ is a fixed vector-clipping radius. Hence $\|\bar{\mathbf{d}}_i\|_2 \le C_{\text{vec}}$. The locally privatized representation is $\widetilde{\mathbf{d}}_i = \bar{\mathbf{d}}_i + \boldsymbol{\eta}_i, \qquad \boldsymbol{\eta}_i \sim \mathcal{N}(0, \tau_{\text{node}}^2 I_K)$. As shown in Theorem 4, the complete clipped representation has $\ell_2$-sensitivity $\Delta_{2,\text{node}} \le \sqrt{2}\, C_{\text{vec}}$. We calibrate $\tau_{\text{node}}$ to the desired $(\epsilon_{\text{node}}, \delta_{\text{node}})$ guarantee using the analytic Gaussian mechanism Balle and Wang (2018). Consequently, the complete $K$-dimensional representation released by one private point satisfies $(\epsilon_{\text{node}}, \delta_{\text{node}})$-LDP.

# 5 EXPERIMENTS

**Datasets and Baselines.** We evaluate our approach on 14 widely used benchmark datasets spanning five categories. (a) *Citation Networks:* Cora, Citeseer, and Pubmed Sen et al. (2008); Acharya and Zhang (2020); Malik et al. (2025), (b) *Co-purchase Networks:* Amazon-Photo Hu et al. (2020), (c) *Co-author and Wiki-page Networks:* Coauthor-CS, Coauthor-Physics Shchur et al. (2018), Chameleon Sun et al. (2023), and Squirrel Pei et al. (2020), (d) *General Networks:* Tolokers (Crowd-sourcing), Amazon-Ratings and Questions (Social) Platonov et al. (2023), (e) *Large Networks: ogbn-arxiv* Wang et al. (2020) , *Reddit* Hamilton et al. (2017) and *ogbn-products* Hu et al. (2020). Additional dataset statistics are provided in Table 6 in Appendix A.3. We compare the proposed CoCo framework with 11 conventional and recent state-of-the-art baselines, which include *FedAvg* McMahan et al. (2017b), *Fed-PUB* Baek et al. (2023), *FedDC* Gao et al. (2022), *FedTGP* Zhang et al. (2024), *MOON* Li et al. (2021), *SCAFFOLD* Karimireddy et al. (2020), *FedGTA* Li et al. (2024b), *FedTAD* Zhu et al. (2024), *FedSage+* Zhang et al. (2021), *AdaFGL* Li et al. (2024a) and *FedIIH* Yu et al. (2025). We have included the detailed discussion about these methods in Sec. 2. For training details and system specifications, see Appendix A.4 and see A.6 respectively.

## 5.1 PERFORMANCE ANALYSIS.

**Node Classification.** We evaluate CoCo for federated node classification against strong FL baselines across 14 benchmark datasets. As shown in Table 1, CoCo achieves the best performance on $12/14$ datasets, demonstrating the effectiveness of its personalized collaboration mechanism. On *Amazon-Photo* and *Reddit*, CoCo attains the second-best performance, remaining highly competitive with only a marginal gap from the top-performing method.

Table 1: Node classification accuracy (%) across federated methods. Δ reports CoCo minus the best non-CoCo method. Results over five runs, including mean ± std., are reported in Table 10. "F-" abbreviates "Fed-".

| Dataset | F-PUB | F-Avg | F-GTA | F-DC | F-TGP | F-TAD | MOON | SCAF. | F-Sage+ | AdaFGL | F-IIH | CoCo | Δ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cora** | 89.59 | 88.37 | 88.36 | 89.60 | 87.38 | 88.36 | 88.51 | 76.66 | 88.91 | 90.32 | 82.66 | **92.04** | +1.72 |
| **Cite.** | 70.74 | 69.24 | 69.15 | 75.17 | 65.16 | 69.33 | 71.09 | 75.26 | 74.50 | 78.40 | 72.58 | **79.87** | +1.47 |
| **Pub.** | 86.47 | 85.55 | 85.44 | 81.81 | 84.48 | 85.44 | 85.64 | 70.14 | 87.93 | 87.44 | 86.58 | **88.10** | +0.17 |
| **Photo** | **92.76** | 91.78 | 91.74 | 91.56 | 78.03 | 91.88 | 91.85 | 84.89 | 90.01 | 89.49 | 90.67 | 92.71 | -0.05 |
| **Squ.** | 44.25 | 39.98 | 38.22 | 35.62 | 42.85 | 40.91 | 40.44 | 35.34 | 49.30 | 47.95 | 28.35 | **49.80** | +0.50 |
| **Tol.** | 78.84 | 77.54 | 77.92 | 78.21 | 79.23 | 78.26 | 77.87 | 77.54 | 78.81 | 78.47 | 69.70 | **79.39** | +0.16 |
| **AR** | 45.00 | 45.04 | 45.06 | 38.81 | 49.08 | 45.11 | 44.54 | 36.81 | 48.13 | 48.26 | 50.24 | **52.21** | +1.97 |
| **Ques.** | 97.01 | 97.01 | 97.01 | 97.01 | 97.04 | 97.01 | 97.01 | 97.01 | 97.03 | 97.02 | 69.75 | **97.08** | +0.04 |
| **CS** | 89.40 | 89.53 | 89.46 | 90.25 | 87.00 | 89.32 | 89.91 | 83.10 | 90.28 | 91.61 | 91.61 | **92.47** | +0.86 |
| **Phys.** | 94.59 | 94.31 | 94.44 | 94.32 | 93.89 | 94.40 | 94.59 | 93.64 | OOM | 95.32 | 95.50 | **95.95** | +0.45 |
| **Cham.** | 58.22 | 55.84 | 55.13 | 49.81 | 56.98 | 55.84 | 56.55 | 47.23 | 60.53 | 56.68 | 42.01 | **61.25** | +0.72 |
| **Arxiv** | 63.47 | 61.35 | 63.25 | 46.62 | 59.07 | 60.29 | 61.20 | 37.25 | 63.49 | OOM | OOM | **64.70** | +1.21 |
| **Reddit** | 92.96 | 93.01 | **94.10** | 91.46 | 90.88 | 92.97 | 92.85 | 86.44 | OOM | OOM | OOM | 93.95 | -0.15 |
| **OG-P** | 82.90 | 82.75 | 85.75 | 71.79 | 80.54 | OOM | 82.71 | 61.85 | OOM | OOM | OOM | **85.91** | +0.16 |

Table 2: Link prediction accuracy.

| Dataset | F-DC | F-Avg | F-Prox | SCAF. | CoCo |
|---|---|---|---|---|---|
| Cora | 67.45 | 71.41 | 71.09 | 65.34 | **73.86** |
| Cite. | 70.51 | 71.64 | 70.76 | 71.52 | **72.73** |
| Pub. | 56.15 | 67.98 | 68.53 | 51.26 | **68.69** |
| Cham. | 67.81 | 74.09 | 74.29 | 62.06 | **75.54** |

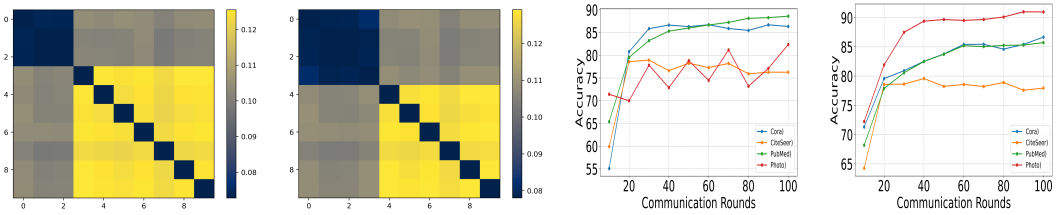
By combining PP-GSI with selective knowledge sharing, CoCo enables each client to retain local specialization while leveraging informative cross-client collaboration. Moreover, CoCo is model-agnostic: when instantiated with different backbones (GCN Kipf and Welling (2016), GraphSAGE Hamilton et al. (2017), and GAT Veličković et al. (2017)), it maintains strong performance trends (Figures 3c, 3d, and 8), indicating robust transferability across GNN architectures. We further study scalability by accelerating the one-time global structure inference via UGC Kataria et al. (2026) coarsening (Sec. 3.4). Table 11 in Appendix B.5 reports (i) the effect of coarsening ratios $\rho = N'/N \in \{0.02, 0.05, 0.1, 0.3, 0.6, 0.7, 0.8, 0.9, 1.0\}$ on representative datasets (Squirrel/Cora/CiteSeer), and (ii) results on large-scale benchmarks where coarsening is essential for feasibility. For Squirrel/Cora/CiteSeer, accuracy remains nearly unchanged even under aggressive reduction, suggesting the coarsened supernode graph preserves the global relations required by CoCo. Using this capability, we scale CoCo to Reddit, ogbn-arxiv and ogbn-products (up to $\approx 2.45$M nodes), (Table 1).

**Link prediction.** Table 2 reports test performance for federated link prediction across standard FL baselines and our proposed CoCo. Across all datasets, CoCo achieves the best results, outperforming the strongest competing baseline. These gains indicate that CoCo more effectively captures cross-client collaboration structure for link prediction, enabling better generalization.

Table 3: **Dynamic CoCo** under distribution shift. **Original**: clean training. **Shifted**: corrupted data with outdated collaboration. **Relearned**: corrupted data with updated collaboration.

| Dataset | $\mathcal{M}_{\text{orig}}$ | $\mathcal{M}_{\text{corr}}$ | $\mathcal{M}_{\text{re}}$ |
|---|---|---|---|
| Cora | 91.26 | 89.05 | 89.90 |
| PubMed | 85.81 | 82.04 | 84.62 |
| Photo | 82.01 | 80.72 | 84.27 |

**Visualization of Collaboration Graphs.** We visualize the learned collaboration graphs for *Citeseer*, *Cora*, *Chameleon* and *Squirrel* in Appendix B.2. Each heatmap shows pairwise collaboration strength (brighter = stronger). Across datasets, we observe clear, data-driven structure: similar clients form dense blocks, while divergent or noisy clients exhibit weakened ties. These visuals confirm that CoCo learns adaptive (non-uniform) collaboration rather than applying a fixed aggregation rule, naturally balancing personalization and sharing.




(a) Noisy Clients: 0,1,2 (b) Noisy Clients: 0,1,2,3 (c) GraphSAGE (d) GAT

Figure 3: (a,b) show noisy-client effects: as noisy clients increase from 3 to 4, their links weaken while clean-client links remain stable, indicating that CoCo isolates corrupted clients and preserves collaboration. (c,d) show accuracy across communication rounds with GraphSAGE and GAT backbones.

### 5.2 DYNAMIC CoCo UNDER NON-STATIONARY CLIENTS.

CoCo infers the collaboration graph one-shot, but in non-stationary settings a fixed graph may become suboptimal. We simulate such shifts by injecting Gaussian noise into randomly selected feature channels of shifted clients. For a shifted client $C_m$, we corrupt selected feature channels $\widetilde{X}_m = X_m + \Gamma_m \odot \Xi_m, \quad (\Xi_m)_{ir} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \tau^2_{\text{shift}}), \quad r = 1, \ldots, d$, where $\Gamma_m \in \{0,1\}^{n_m \times d}$ is a binary mask selecting the perturbed feature channels. This changes the anchor-based distance signals, leading to weaker inferred connections involving shifted clients while largely preserving those among unshifted clients (Fig. 3a, Fig. 3b, Appendix A.5).

To isolate the effect of updating the collaboration structure, both post-shift evaluations start from the same pre-shift model checkpoint and use the same number of additional training rounds and identical optimization settings. For $\mathcal{M}_{\text{corr}}$, training continues on the shifted data while retaining the pre-shift collaboration graph. For $\mathcal{M}_{\text{re}}$, we first rerun PP-GSI and reconstruct the collaboration graph from the shifted data, after which training proceeds under the same protocol. Thus, the difference between $\mathcal{M}_{\text{corr}}$ and $\mathcal{M}_{\text{re}}$ isolates the effect of re-estimating the collaboration graph rather than additional optimization. Note that $\mathcal{M}_{\text{orig}}$ in Table 3 corresponds to the pre-shift checkpoint of this dynamic experiment, whereas Table 1 reports CoCo after full training. The absolute values are therefore not expected to coincide. Table 3 shows that performance degrades when the shifted data are used with the stale collaboration graph and improves after the collaboration structure is re-estimated.

### 5.3 PRIVACY AND FIDELITY.

**Varying anchor count.** We vary the number of anchors $K$ around the ambient feature dimension $d$ and evaluate F-score and Frobenius reconstruction error (Fig. 4a, 4b, details in Appendix C.1). Even for $K < d$ (e.g., $K = d - 100$), F-score steadily improves, showing strong structural fidelity with limited anchors. As $K \to d$ or $K > d$, $X_{\text{frob}} \to 0$, indicating reduced reconstruction ambiguity. Thus, $K < d$ provides protection against unique reconstruction while retaining useful global structure, LDP complements this guarantee against broader information leakage.

**Differential Privacy Results.** We evaluate CoCo under two DP mechanisms operating at different stages of the pipeline: *(1) CoCo under approximate node-level LDP* applied to the shared private-to-anchor distance representation (details in Appendix H.1). In Table 4, we vary $\epsilon_{\text{node}}$ while fixing $\delta_{\text{node}} = 10^{-6}$. Across Cora, CiteSeer, and Amazon-Ratings (AR), performance remains stable under increasingly strong perturbation. We also study reconstruction leakage across varying anchor counts in Fig. 10, node-level LDP maintains non-negligible reconstruction error even as $K$ approaches or exceeds the feature dimension, while preserving the downstream utility reported in Table 4. We further evaluate empirical leakage under membership, property, attribute, approximate-reconstruction, and link-inference attacks in Appendix B.8.

Table 4: Accuracy (%) for different LDP budgets.

| Setting | Cora | Cite. | AR |
|---|---|---|---|
| **No DP** | **92.04** | **79.87** | **52.21** |
| $\epsilon$=16 | 90.96 | 78.35 | 46.50 |
| $\epsilon$=8 | 90.68 | 78.36 | 46.47 |
| $\epsilon$=4 | 90.61 | 78.25 | 46.45 |
| $\epsilon$=2 | 90.58 | 78.23 | 46.42 |
| $\epsilon$=0.1 | 90.50 | 78.25 | 46.37 |

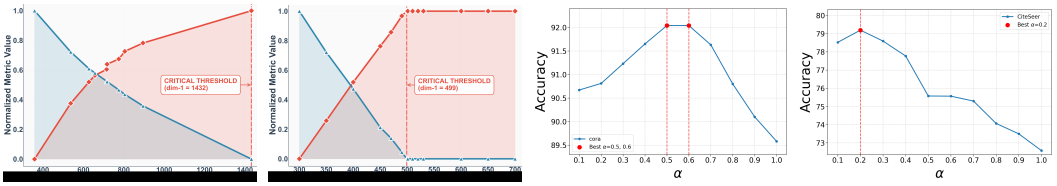
*(2) Model-level DP (OpenFGL).* We additionally evaluate CoCo under standard model-level DP using the OpenFGL framework Li et al. (2024c). In this setting, DP is enforced during training via gradient clipping and noise injection. We compare against DP-enabled baselines including SCAFFOLD, MOON, and FedAvg, with results reported in Appendix B.9 in Table 15. We sweep the privacy budget $\epsilon_{model} \in \{1, 5, 10\}$ and report node classification accuracy on *PubMed* and *Amazon-Ratings*. Across all datasets and privacy levels, CoCo consistently outperforms competing methods. While stronger privacy (smaller $\epsilon$) introduces additional noise, the resulting performance degradation remains limited, indicating a favorable privacy–utility trade-off.

### 5.4 ABLATION STUDY.

*1) Varying Communication Rounds.* Figure 3c and 3d (additional results in Figure 8) show accuracy across communication rounds. Since the collaboration graph is inferred once and reused, CoCo avoids repeated inter-client similarity estimation during training. Accuracy reaches near-peak values within few rounds and then saturates, indicating rapid and stable training behavior across different GNN backbones. This efficiency comes from one-shot structure inference and collaboration graph learning, enabling effective aggregation with fewer rounds.

*2) Varying Number of Clients.* Figure 5 reports the test accuracy of CoCo as the number of clients increases (5, 10, 15, 20, 25) on *Cora*, *Squirrel*, and *Chameleon*. Accuracy drops slightly due to

stronger data fragmentation and fewer samples per client, which is expected in FL. Importantly, the degradation is modest, indicating that CoCo remains stable under larger client counts. More baseline and client-scaling trends are deferred to Appendix B.4.




(a) Cora (b) Pubmed (c) Cora (d) CiteSeer

Figure 4: (a,b) show anchor-count effects under non-IID unbalanced splits. The red line marks $d-1$ (1432 for Cora, 499 for Pubmed), beyond which exact recovery may occur, for Cora it lies outside the range, giving monotonic gains. Trends show the fidelity–privacy trade-off. (c,d) show the effect of $\alpha$ on Cora and CiteSeer.

*3) Effect of Varying Personalization ($\alpha$).* The parameter $\alpha$ balances local learning and collaboration: lower values increase inter-client sharing, while higher values favor local models. Figure 4c and 4d show that moderate values ($\alpha = 0.5/0.6$ for Cora, $\alpha = 0.2$ for CiteSeer) perform best, indicating that CoCo adapts to different heterogeneity levels. Purely local or fully collaborative or global extremes reduce generalization.

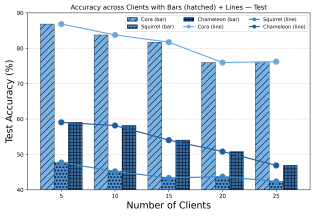



Figure 5: Test accuracy of CoCo with varying clients.

*4) Anchor generation ablation.* Table 12 in Appendix B.6 evaluates CoCo under different anchor constructions. *Syn-Gaussian* samples anchors i.i.d. as $a_j \sim \mathcal{N}(0, I_d)$, while *SynOrtho-Gaussian* additionally orthogonalizes the sampled anchor directions to improve geometric conditioning. *In-Data ($< d$)* draws in-distribution anchors from the dataset with $K < d$ where $K$ is the number of anchors. CoCo is effective across all choices, with *In-Data* performing best, indicating that in-distribution anchors are most informative for PP-GSI. Notably, CoCo remains competitive with non-In-Data anchors that do not require client-private samples, supporting the use of public or governance-approved domain-compatible references (Appendix A.2).

### 5.5 Discussion.

**Data Heterogeneity in FGL.** A key challenge in FGL is *data heterogeneity* across clients. Following prior work Li et al. (2024c), heterogeneity in FGL is commonly characterized along three dimensions: (i) FEATURE SKEW (differences in node-attribute distributions), (ii) LABEL SKEW (differences in label distributions), and (iii) TOPOLOGY SKEW (differences in local graph structure). These forms of heterogeneity can affect optimization and generalization, motivating controlled evaluation across complementary partitioning regimes. Further details are provided in Appendix C.2.

Table 5: Accuracy under different partitioning schemes.

| Partition | Pub. | Cora | Cite. |
|---|---|---|---|
| Label Skew | 90.02 | 89.18 | 74.78 |
| Louvain+ | 88.10 | 92.04 | 79.87 |
| Metis+ | 89.95 | 91.25 | 78.03 |
| Louvain | 89.31 | 92.72 | 77.90 |
| Metis | 88.78 | 93.13 | 78.01 |

*Partitioning strategies.* We adopt OpenFGL simulation protocols Li et al. (2024c) (Appendix C.2) and evaluate label-skew and community-based splits (Louvain/Metis) along with their enhanced variants (Louvain+/Metis+). These standardized partitions are used as controlled and reproducible stress tests for distinct forms of client heterogeneity, rather than as exact models of a particular deployment scenario. Table 5 reports results on PubMed, Cora, and CiteSeer. CoCo remains consistently strong across different splits, with community-based splits generally benefiting from greater preservation of graph structure. Overall, the results indicate robustness to both label and structural heterogeneity.

## 6 Conclusion

In federated graph learning, client graphs are often non-IID. We therefore adopt a personalized federated graph learning setting, where a key challenge is balancing individual utility with collaborative benefit. CoCo addresses this by learning a collaboration graph that weights aggregation toward similar, beneficial peers, improving effective local homogeneity. Experiments across diverse graph datasets show CoCo outperforms baselines on different downstream tasks, demonstrating effective adaptive personalization.

## AI USE STATEMENT

We used generative AI tools, including large language models (LLMs), for language-editing assistance such as paraphrasing, grammar correction, improving clarity, and checking notation and terminology consistency across the manuscript. All AI-assisted edits were reviewed by the authors for correctness, consistency, and fidelity to the intended technical content. The authors take full responsibility for the final content of this work, including all text, claims, results, and artifacts.

## ETHICS STATEMENT

This work does not involve human-subject experiments, collection of personally identifiable information, or release of new sensitive datasets. Our experiments are conducted on established benchmark datasets and are intended to evaluate federated graph learning methodology under controlled settings.

## SOCIETAL IMPACT

CoCo can enable privacy-preserving graph learning in practical settings where data cannot be centralized due to legal, ethical, or institutional constraints. This is especially important in critical domains such as healthcare, finance, public services, and biomedical research, where organizations may need to train models collaboratively while keeping sensitive user, patient, or institutional data local. By inferring a collaboration graph from privacy-preserving signals and supporting decentralized personalized learning, CoCo can help improve model utility without requiring raw data sharing. At the same time, deployment in high-stakes settings should include careful governance, consent, auditing, and privacy-risk assessment to avoid unintended leakage or biased decision-making.

## REPRODUCIBILITY STATEMENT

We provide the implementation of CoCo, including the code required to reproduce the main experimental results, at https://anonymous.4open.science/r/CoCo-C5B3/. The main paper describes the proposed methodology and experimental protocol, while the appendix provides additional implementation details, dataset statistics, training configurations, runtime analysis, privacy settings, and complete proofs of the theoretical results. Together, these materials are intended to support reproducibility of both the empirical and theoretical findings reported in this work.

## REFERENCES

Han Xie, Jing Ma, Li Xiong, and Carl Yang. Federated graph classification over non-iid graphs. Advances in neural information processing systems, 34:18839–18852, 2021.

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

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine, 37(3):50–60, 2020.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. Advances in neural information processing systems, 33:19586–19597, 2020.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. arXiv preprint arXiv:2002.10619, 2020.

Nikita Kotelevskii, Maxime Vono, Alain Durmus, and Eric Moulines. Fedpop: A bayesian approach for personalised federated learning. Advances in Neural Information Processing Systems, 35: 8687–8701, 2022.

Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. Pyramidfl: A fine-grained client selection framework for efficient federated learning. In Proceedings of the 28th annual international conference on mobile computing and networking, pages 158–171, 2022.

Yongheng Deng, Feng Lyu, Ju Ren, Huaqing Wu, Yuezhi Zhou, Yaoxue Zhang, and Xuemin Shen. Auction: Automated and quality-aware client selection framework for efficient federated learning. IEEE Transactions on Parallel and Distributed Systems, 33(8):1996–2009, 2021.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. IEEE transactions on neural networks and learning systems, 34(12):9587–9603, 2022.

Youssef Allouah, Abdellah El Mrini, Rachid Guerraoui, Nirupam Gupta, and Rafael Pinot. Fine-tuning personalization in federated learning to mitigate adversarial clients. Advances in Neural Information Processing Systems, 37:100816–100844, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics, pages 1273–1282. Pmlr, 2017a.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10708–10717. IEEE, 2021.

Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. Personalized federated learning with inferred collaboration graphs. In International Conference on Machine Learning, pages 39801–39817. PMLR, 2023.

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

Wentao Yu, Shuo Chen, Yongxin Tong, Tianlong Gu, and Chen Gong. Modeling inter-intra heterogeneity for graph federated learning. In Proceedings of the AAAI conference on artificial intelligence, volume 39, pages 22236–22244, 2025.

Xunkai Li, Zhengyu Wu, Wentao Zhang, Yinlin Zhu, Rong-Hua Li, and Guoren Wang. Fedgta: Topology-aware averaging for federated graph learning. arXiv preprint arXiv:2401.11755, 2024b.

Jinheon Baek, Wonyong Jeong, Jiongdao Jin, Jaehong Yoon, and Sung Ju Hwang. Personalized subgraph federated learning. In International conference on machine learning, pages 1396–1415. PMLR, 2023.

Yinlin Zhu, Xunkai Li, Zhengyu Wu, Di Wu, Miao Hu, and Rong-Hua Li. Fedtad: Topology-aware data-free knowledge distillation for subgraph federated learning. arXiv preprint arXiv:2404.14061, 2024.

Jianqing Zhang, Yang Liu, Yang Hua, and Jian Cao. Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In Proceedings of the AAAI conference on artificial intelligence, volume 38, pages 16768–16776, 2024.

Mohit Kataria, Nikita Malik, Sandeep Kumar, et al. Fast and scalable hashing-based universal graph coarsening. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2026.

Mohit Kataria, Sandeep Kumar, et al. Ugc: Universal graph coarsening. Advances in Neural Information Processing Systems, 37:63057–63081, 2024.

Thu LN Nguyen and Yoan Shin. Matrix completion optimization for localization in wireless sensor networks for intelligent iot. Sensors, 16(5):722, 2016.

Nikita Malik, Nimesh Agrawal, and Sandeep Kumar. Ppda: A privacy preserving framework for distributed graph learning. In International Conference on Neural Information Processing, pages 308–331. Springer, 2024.

Moses A Koledoye, Tullio Facchinetti, and Luis Almeida. Mds-based localization with known anchor locations and missing tag-to-tag distances. In 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pages 1–4. IEEE, 2017.

Mu Zhou, Yaohua Li, Yiwen Wang, Qiaolin Pu, Xiaolong Yang, and Wei Nie. Device-to-device cooperative positioning via matrix completion and anchor selection. IEEE Internet of Things Journal, 9(7):5461–5473, 2021.

Carmelo Di Franco, Enrico Bini, Mauro Marinoni, and Giorgio C Buttazzo. Multidimensional scaling localization with anchors. In 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pages 49–54. IEEE, 2017.

Vassilis Kalofolias. How to learn a graph from smooth signals. In Artificial intelligence and statistics, pages 920–929. PMLR, 2016.

Samuel Lichtenberg and Abiy Tasissa. Localization from structured distance matrices via low-rank matrix recovery. IEEE Transactions on Information Theory, 2024.

Guangcan Liu and Shuicheng Yan. Latent low-rank representation for subspace segmentation and feature extraction. In 2011 International Conference on Computer Vision, pages 1615–1622, 2011. doi: 10.1109/ICCV.2011.6126422.

Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse and low-rank subspace clustering. IEEE Journal of Selected Topics in Signal Processing, 9(4):691–701, 2015.

Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 394–403. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/balle18a.html.

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

Deepak Bhaskar Acharya and Huaming Zhang. Feature selection and extraction for graph neural networks. In Proceedings of the 2020 ACM southeast conference, pages 252–255, 2020.

Nikita Malik, Rahul Gupta, and Sandeep Kumar. Hyperdefender: A robust framework for hyperbolic gnns. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 19396–19404, 2025.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems, 33:22118–22133, 2020.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. ArXiv preprint, 2018.

Junshu Sun, Shuhui Wang, Xinzhe Han, Zhe Xue, and Qingming Huang. All in a row: Compressed convolution networks for graphs. In International Conference on Machine Learning, pages 33061–33076. PMLR, 2023.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? arXiv preprint arXiv:2302.11640, 2023.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. Quantitative Science Studies, 1(1): 396–413, 2020.

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

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

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10112–10121, 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, pages 5132–5143. PMLR, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

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

Xunkai Li, Yinlin Zhu, Boyang Pang, Guochen Yan, Yeyu Yan, Zening Li, Zhengyu Wu, Wentao Zhang, Rong-Hua Li, and Guoren Wang. Openfgl: A comprehensive benchmark for federated graph learning. arXiv preprint arXiv:2408.16288, 2024c.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 7370–7377, 2019.

Yuzhong Peng, Yanmei Lin, Xiao-Yuan Jing, Hao Zhang, Yiran Huang, and Guang Sheng Luo. Enhanced graph isomorphism network for molecular admet properties prediction. Ieee Access, 8: 168344–168360, 2020.

Andreas Loukas. Graph reduction with spectral and cut guarantees. Journal of Machine Learning Research, 20(116):1–42, 2019.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (SP), pages 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. Proceedings on Privacy Enhancing Technologies, 2022(4):528–551, 2022. doi: 10.56553/popets-2022-0121.

Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. ML-Doctor: Holistic risk assessment of inference attacks against machine learning models. In 31st USENIX Security Symposium (USENIX Security 22), pages 4525–4542. USENIX Association, 2022.

Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. Stealing links from graph neural networks. In 30th USENIX Security Symposium (USENIX Security 21), pages 2669–2686. USENIX Association, 2021.

Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega. Graph learning from data under structural and laplacian constraints. arXiv preprint arXiv:1611.05181, 2016.

Usman A Khan, Soummya Kar, and José MF Moura. Distributed sensor localization in random environments using minimal number of anchor nodes. IEEE Transactions on Signal Processing, 57(5):2000–2016, 2009.

John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Minimax optimal procedures for locally private estimation. Journal of the American Statistical Association, 113(521):182–201, 2018.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE transactions on neural networks and learning systems, 32(8):3710–3722, 2020.

Sina Sajadmanesh and Daniel Gatica-Perez. Locally private graph neural networks. In Proceedings of the 2021 ACM SIGSAC conference on computer and communications security, pages 2130–2145, 2021.

Arthur Sard. The measure of the critical values of differentiable maps. Bulletin of the American Mathematical Society, 48:883–890, 1942.

Thomas M Cover. Elements of information theory. John Wiley & Sons, 1999.

David Tse and Pramod Viswanath. Fundamentals of wireless communication. Cambridge university press, 2005.

# Appendix


| Section | Appendix Contents | Page |
|---|---|---|
| A | **Methodology and Dataset** | **14** |
| A.1 | **Background and Related Work** | 14 |
| A.2 | **Anchor Generation** | 15 |
| A.3 | **Dataset Statistics** | 16 |
| A.4 | **Training Details** | 16 |
| A.5 | **Robust and Dynamic CoCo Results** | 17 |
| A.6 | **System Specifications** | 17 |
| A.7 | **CoCo Algorithm** | 17 |
| A.8 | **Run-Time Analysis** | 17 |
| B | **Additional Results** | **19** |
| B.2 | **Visualization of Collaboration Graphs** | 19 |
| B.3 | **Results for Varying Communication Rounds** | 20 |
| B.4 | **Effect of Varying Client Count on Baseline Methods** | 20 |
| B.5 | **Varying Coarsening Ratio for Scalable CoCo** | 20 |
| B.6 | **Anchor generation ablation results** | 20 |
| B.7 | **Additional DP results** | 21 |
| C | **Additional Metrics and Implementation Details** | **23** |
| C.1 | **F-score and Frobenius Error** | 23 |
| C.2 | **Partition Schemes** | 23 |
| D | **Theorem and Lemma Proofs** | **24** |
| D.1 | **Stability of Learned Global Structure** | 24 |
| D.2 | **Client-Coarsening Stability** | 26 |
| D.3 | **Graph-Smoothing Property of CoCo Collaboration** | 27 |
| D.4 | **Non-Identifiability of Anchor-Distance Representations** | 27 |
| D.5 | **Sensitivity of Distance-Based Representation** | 28 |
| F | **Privacy Model and Federated Information Sharing** | **29** |
| F | **More Theoretical Guarantees with Proofs** | **30** |
| | **Theorem: Deterministic Geometric Non-identifiability** | 31 |
| G | **Scaling CoCo to Large Graphs** | **33** |
| H | **Implementation Details for Differential Privacy Experiments** | **35** |
| H.1 | **Node-Level Approximate LDP** | 35 |
| H.2 | **Model-Level DP** | 35 |


## A Methodology and Dataset

This section provides background definitions, anchor construction details, dataset statistics, training settings, robustness setup, system specifications, algorithm, and runtime analysis.

### A.1 Background and Related Work

**Graphs and GNNs.** The (unnormalized) Graph Laplacian is then given by $L = D - A$ Kipf and Welling (2016). A matrix $L \in \mathbb{R}^{N\times N}$ is a valid Laplacian if it belongs to the set:

$$\mathcal{S}_L = \left\{ L \in \mathbb{R}^{N\times N} \;\middle|\; L_{ij} = L_{ji} \le 0,\ \forall i \neq j,\ L_{ii} = -\sum_{j\neq i} L_{ij} \right\} \tag{5}$$

GNNs Yao et al. (2019); Veličković et al. (2017); Peng et al. (2020); Hamilton et al. (2017); Malik et al. (2025) have become a central paradigm for learning from this graph-structured data. They jointly exploit node features and graph topology to produce expressive embeddings. Also, they are

widely used for semi-supervised node-, link-, and graph-level learning across domains. In this work, we mainly consider the node classification task. The goal in node classification is to assign each node $v_i \in V$ to one of $C_{\text{cls}}$ categories. Given a partially labeled graph, a GNN $f_\theta$ learns to predict labels $\mathcal{Y}_L$ for a subset of labeled nodes $\mathcal{V}_L \subset V$ and generalizes to unlabeled nodes by exploiting structural information. The objective is formulated as:

$$\min_{\theta} \mathcal{L}(\theta, A, X, \mathcal{Y}_L) = \sum_{v_i \in \mathcal{V}_L} \ell(f_\theta(X, A)_i, y_i)\,. \tag{6}$$

where $f_\theta(X, A)_i$ is the predicted label distribution for node $v_i$, $y_i$ is the true label, and $\ell(\cdot, \cdot)$ is typically the cross-entropy loss. This objective enforces discriminative embeddings that leverage both node attributes and graph connectivity.

**Graph Coarsening (GC).**

Early works on GC focused on preserving the spectral properties of the graph Laplacian Loukas (2019). More recent approaches have extended this idea to feature-aware settings, where both structural and attribute similarity guide node aggregation. Adaptive methods, such as UGC Kataria et al. (2024; 2026), further introduce randomized hashing strategies to merge nodes probabilistically, enabling scalable coarsening. Overall, GC provides a principled mechanism for compressing large-scale graphs into compact surrogates, significantly reducing computational costs while retaining essential structural, spectral, and attribute information for downstream graph learning.

### A.2 Anchor Generation

The design and distribution of anchors play a pivotal role in governing both privacy and reconstruction quality. Anchors act as shared reference points accessible to all participating clients, enabling cross-client structural alignment without exposing raw data. To preserve privacy, anchor data are typically either generated synthetically or drawn from open, non-sensitive datasets. Proper scheduling and management of anchors critically influence the overall accuracy, stability, and scalability of the framework. Two key considerations guide the anchor generation strategy:

**Number of anchors.** The number of anchors $K$ directly affects both privacy and fidelity. A very small $K$ may fail to capture sufficient structural information for accurate reconstruction, whereas an excessively large $K$ increases computational cost and can reduce the geometric ambiguity of the private-point representation. Hence, $K$ must be chosen to balance structure preservation and privacy guarantees.

**Selection criteria.** The effectiveness of the anchors also depends on how they are selected. Anchors drawn from the same underlying distribution as the private data typically provide better alignment and reconstruction fidelity than purely random anchors. For instance, in patient-similarity or social-network applications, relevant distributional characteristics such as feature correlations and population structure should ideally be reflected in the reference anchors.

In our default experimental setting, the anchors are sampled from a held-out subset of the benchmark distribution that is excluded from all private client partitions, thereby providing domain-compatible reference points without using client-private records. We additionally evaluate two synthetic anchor-generation strategies (Table 12) for settings in which such in-distribution reference data are unavailable: (i) *Syn-Gaussian*, where anchors are sampled i.i.d. from $\mathcal{N}(0, I_d)$, and (ii) *SynOrtho-Gaussian*, where the sampled anchor directions are additionally orthogonalized to improve geometric conditioning.

**Practical considerations.** In our experiments, subsets of the original data are used as anchors solely for empirical evaluation, as it is important that anchors follow the same distribution as the underlying data. In real-world deployments, the nature of the task suffices to determine the appropriate anchor type. For example, in distributed healthcare, hospitals collaborating to build a patient similarity network for diagnosis or risk prediction can contribute a small, non-sensitive subset of patient data as reference samples. The central server can then aggregate these samples to form a shared anchor set and redistribute it to all participating institutions. Analogously, financial organizations cooperating on fraud detection may provide anonymized or representative transaction samples to serve as anchors. Alternatively, publicly available datasets can be directly employed as privacy-preserving anchors when domain-compatible data exist.

**Deployment checklist (anchors).** When anchors are contributed by participants, we assume they satisfy: (i) consent and governance approval, (ii) removal of direct identifiers, (iii) no rare/unique individuals or extreme outliers, and (iv) bounded contribution size per institution. When anchors are public or synthetic, we recommend measuring domain compatibility via a lightweight proxy (e.g., feature mean/covariance similarity) since out-of-domain anchors can degrade PP-GSI alignment.

Unless otherwise stated, we use in-distribution reference anchors drawn from a held-out reference subset that is excluded from all client partitions and treated as non-private shared data.

### A.3 Dataset Statistics.

We evaluate our approach on 14 widely used benchmark datasets spanning different categories. Table 6 contains details about the number of nodes, edges, features, classes and Train/Val/Test split.

Table 6: The statistical information of the experimental datasets.

| Dataset | Nodes | Edges | Features | Classes | Train/Val/Test | Description |
|---|---|---|---|---|---|---|
| Cora | 2,708 | 5,429 | 1,433 | 7 | 80%/10%/10% | Citation Network |
| CiteSeer (Cite.) | 3,327 | 4,732 | 3,703 | 6 | 80%/10%/10% | Citation Network |
| PubMed (Pub.) | 19,717 | 44,338 | 500 | 3 | 80%/10%/10% | Citation Network |
| Amazon-Photo (Photo) | 7,487 | 119,043 | 745 | 8 | 80%/10%/10% | Co-purchase Network |
| Co-author CS (CS) | 18,333 | 81,894 | 6,805 | 15 | 80%/10%/10% | Co-author Network |
| Co-author Physics (Phy.) | 34,493 | 247,962 | 8,415 | 5 | 80%/10%/10% | Co-author Network |
| Chameleon (Cham.) | 2,277 | 36,101 | 2,325 | 5 | 80%/10%/10% | Wiki-page Network |
| Squirrel (Sqi.) | 5,201 | 216,933 | 2,089 | 5 | 80%/10%/10% | Wiki-page Network |
| Tolokers (Tol.) | 11,758 | 519,000 | 10 | 2 | 80%/10%/10% | Crowd-sourcing Network |
| Amazon-ratings (AR) | 24,492 | 93,050 | 300 | 5 | 80%/10%/10% | Rating Network |
| Questions (Ques.) | 48,921 | 153,540 | 301 | 2 | 80%/10%/10% | Social Network |
| Ogbn-arxiv (Arxiv) | 169,343 | 1,166,243 | 128 | 40 | 80%/10%/10% | - |
| Reddit | 232,965 | 114,615,892 | 602 | 41 | 80%/10%/10% | - |
| Ogbn-products (OG-P) | 2,449,029 | 61,859,140 | 100 | 47 | 80%/10%/10% | - |

### A.4 Training Details.

**Experimental Setup.** Unless otherwise stated, each client trains a GCN-based local model for *100 communication rounds*. Optimization is performed using *Adam* with learning rate $0.01$, L2 regularization $5 \times 10^{-4}$, dropout $0.5$, and batch size $128$. To ensure scalability, we control the graph size using a *coarsening ratio* $\rho = N'/N$ (fraction of nodes retained):

Table 7: coarsening ratio $\rho = N'/N$ used to control scalability across different graph sizes.

| Graph Size (Nodes) | coarsening ratio $\rho = N'/N$. |
|---|---|
| $<$5K | No coarsening |
| 5K–10K | 0.3 |
| 10K–100K | 0.1 |
| $>$100K | 0.02 |

This setup balances computational efficiency and structural fidelity, and is used for results reported in Table 1.

**Scaling to Large Graphs (Table 11).** To evaluate scalability, coarsening experiments are conducted on *Squirrel*, *Cora*, and *CiteSeer* datasets using *10 clients*. Hyperparameters remain consistent with the default setup (learning rate $0.01$, dropout $0.5$, L2 $5 \times 10^{-4}$).

**Heterogeneity Analysis (Table 5).** We study the effect of different partitioning strategies using the setups described in Section C.2. Experiments are performed on *PubMed*, *Cora*, and *CiteSeer*, while maintaining the same configuration as the default setup used in Table 1.

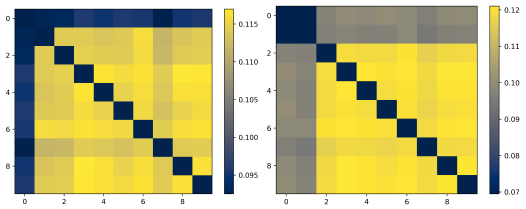
**Varying Number of Clients (Figure 5).** We partition *Cora* (2708 nodes), *Squirrel* (5201 nodes), and *Chameleon* (2277 nodes) into $\{5, 10, 15, 20, 25\}$ clients. Training follows the default configuration (Table 1), and all baselines adhere to the same protocol.

**Varying Communication Rounds (Figure 3c, 3d and 8).** Experiments are conducted on *Cora*, *CiteSeer*, *PubMed*, and *Amazon-Photo*. We record mean validation accuracy every 10 rounds using four backbone architectures: 1) Two-layer GCN, 2) Two-layer GraphSAGE, 3) Two-layer GAT (8 heads in first layer) and 4) Two-layer GAT (1 head in first layer). All models use hidden size $= 8$ and follow the default training setup.

**Robustness to Noisy Clients (Section 5.2 and 6).** We evaluate robustness on the *Cora* dataset by injecting additive Gaussian noise into node features of the first $n$ clients. We then analyze its effect on the PP-GSI induced collaboration matrix and overall accuracy. The configuration matches the default setup in Table 1.

### A.5 Robust and Dynamic CoCo results.

We present additional results for showing the effect of noisy clients on collaboration structure in Figure 6.



(a) Noisy client set: $\{0\}$ (b) Noisy client set: $\{0, 1\}$

Figure 6: Effect of noisy clients on collaboration structure. As the number of noisy clients increases from (a) 1 to (b) 2, their inter-client links weaken (darker regions), while connections among clean clients remain stable. This shows that CoCo effectively isolates corrupted participants and preserves global collaboration fidelity.

**Effect of Personalization (Figure 4c and 4d).** We vary the aggregation personalization factor $\alpha$ in the range $[0.1, 1.0]$, while keeping all other settings identical to the default configuration used in Table 1.

### A.6 System Specifications.

All the experiments conducted for this work were performed on a server with a 72-core Intel(R) Xeon(R) Platinum 8360Y CPU, 1 TB RAM, and an NVIDIA RTX A6000 GPU (48 GB), under Ubuntu 20.04.1.

### A.7 Algorithm.

Our framework has three core stages: (i) privacy-preserving global structure inference, (ii) collaboration-graph construction through coarsening, and (iii) personalized federated graph learning guided by this graph. Algorithm 1 presents the scalable implementation used for large graphs, where local graph coarsening is performed before PP-GSI. For datasets where this scalability step is unnecessary, we set $\Pi_m = I_{n_m}$ and $Z_m = X_m$, so that $n'_m = n_m$ and $N' = N$ and the algorithm reduces to the node-level formulation.

### A.8 Run-Time Analysis

Let $N = \sum_{m=1}^{M} n_m$ be the total number of private points, $K$ the number of public anchors, $d$ the feature dimension, and $T_{\mathrm{MDS}}$ the number of anchored-SMACOF iterations. In PP-GSI, each client computes and transmits only its private-to-anchor distance block, giving $O(NKd)$ aggregate client-side computation and $O(NK)$ communication. At the server, reconstruction is performed independently for each client using the common anchor set, for client $C_m$, the anchored-MDS computation requires $O(T_{\mathrm{MDS}} n_m K d)$ work, yielding $O(T_{\mathrm{MDS}} N K d)$ aggregate reconstruction cost. Since these client-wise reconstruction problems are independent, our implementation parallelizes them, reducing the ideal wall-clock reconstruction cost to $O(T_{\mathrm{MDS}} n_{\max} K d)$, where $n_{\max} = \max_m n_m$. After reconstruction, the client embeddings are concatenated and used by a graph structure learning module to infer the global graph. Since CoCo is agnostic to the particular graph-learning method, we denote this cost by $\mathcal{C}_{\mathrm{GL}}(N, d)$. Hence, including the one-time anchor-anchor distance computation, the total structure-inference cost is

$$O\big(K^2 d + NKd + T_{\mathrm{MDS}} NKd\big) + \mathcal{C}_{\mathrm{GL}}(N, d).$$

For the scalable coarsened variant, replacing $N$ by $N' \ll N$ gives

$$O\big(K^2 d + N'Kd + T_{\mathrm{MDS}} N'Kd\big) + \mathcal{C}_{\mathrm{GL}}(N', d),$$

**Algorithm 1** Scalable implementation of CoCo with optional local coarsening.

**Require:** $\mathcal{C} = \{C_1, \ldots, C_M\}$, $X_m \in \mathbb{R}^{n_m \times d}$, $X_A \in \mathbb{R}^{K \times d}$, $\alpha \in [0,1]$, $E$, $T$
**Ensure:** $\{\theta_1, \ldots, \theta_M\}$
**Client $C_m$ (in parallel)**
$\Pi_m \in \{0,1\}^{n'_m \times n_m}$ ▷ Coarsening matrix
$D_{\Pi_m} \leftarrow \mathrm{Diag}(\Pi_m \mathbf{1})$
$Z_m \leftarrow D_{\Pi_m}^{-1} \Pi_m X_m \in \mathbb{R}^{n'_m \times d}$
**for** $u = 1$ to $n'_m$ **do**
  **for** $j = 1$ to $K$ **do**
    $d(z_{m,u}, a_j) \leftarrow \|z_{m,u} - a_j\|_2$
  **end for**
**end for**
Send $\{d(z_{m,u}, a_j)\}_{u=1,j=1}^{n'_m, K}$ and $n'_m$ to the server
**Server (global structure on supernodes)**
Complete distances $\Rightarrow \widehat{X}_{\mathrm{SN}} \in \mathbb{R}^{N' \times d}$, $\widehat{L}'_G$
Construct $S' \in \{0,1\}^{N' \times M}$ ▷ Supernode–client membership
$P' \leftarrow (S'^{\top} S')^{-1} S'^{\top}$ ▷ Normalized client projection
$\widehat{L}'_C \leftarrow P' \widehat{L}'_G P'^{\top}$
$\bar{A}'_C \leftarrow \mathrm{Diag}\big(\mathrm{diag}(\widehat{L}'_C)\big) - \widehat{L}'_C$
$A'_C \leftarrow \left[\frac{\bar{A}'_C + (\bar{A}'_C)^{\top}}{2}\right]_+$
$D'_C \leftarrow \mathrm{Diag}(A'_C \mathbf{1})$
$W \leftarrow {D'_C}^{-1/2} A'_C {D'_C}^{-1/2}$
Broadcast $W$
**Personalized Training**
Initialize $\{\theta_m^{(0)}\}_{m=1}^M$
**for** $t = 0$ to $T-1$ **do**
  **for** each $C_m \in \mathcal{C}$ **in parallel do**
    Perform local SGD for $E$ steps: $\widetilde{\theta}_m^{(t+1)}$
    Send $\widetilde{\theta}_m^{(t+1)}$ to $\{\ell : W_{\ell m} > 0\}$
    Receive $\{\widetilde{\theta}_\ell^{(t+1)} : W_{m\ell} > 0\}$
    $\theta_m^{(t+1)} \leftarrow \alpha \widetilde{\theta}_m^{(t+1)} + (1-\alpha) \sum_{\ell \neq m} W_{m\ell} \widetilde{\theta}_\ell^{(t+1)}$
  **end for**
**end for**
**return** $\{\theta_m^{(T)}\}_{m=1}^M$

Table 8: Wall-clock communication time (seconds) per round.

| Data | CoCo | FedPub | FedGTA | FedDC | FedTGP | MOON | SCAFFOLD | FedTAD | FedSage+ |
|---|---|---|---|---|---|---|---|---|---|
| Cora | 1.5 | 24.0 | 4.1 | 1.8 | 6.0 | 2.7 | 1.5 | 23.4 | 14.2 |
| CiteSeer | 1.2 | 4.0 | 4.8 | 1.9 | 5.2 | 2.8 | 1.6 | 22.3 | 12.9 |
| PubMed | 1.2 | 4.2 | 3.7 | 1.8 | 4.0 | 3.2 | 2.1 | 15.0 | 35.3 |
| Photo | 1.2 | 3.1 | 3.7 | 2.0 | 5.5 | 2.7 | 1.3 | 25.1 | 166.3 |

with ideal parallel reconstruction time $O(T_{\mathrm{MDS}} n'_{\max} K d)$.

To evaluate the practical training overhead, Table 8 reports the wall-clock communication time per round. Since CoCo does not iteratively re-estimate client similarities or update the collaboration graph during training, each subsequent round requires only local optimization followed by fixed-weight model aggregation. Thus, after the one-time PP-GSI stage, the aggregation performed by CoCo has the same computational form as standard weighted federated averaging while producing personalized client models through the learned collaboration weights.

Across the evaluated datasets, CoCo remains comparable to the lightest federated-training baselines. In contrast, methods that perform additional personalization or auxiliary computations during training can incur substantially larger per-round costs. This highlights an important property of CoCo: personalization is determined once from the inferred collaboration structure rather than repeatedly recomputed from training-dependent signals.

Table 9: One-time PP-GSI preprocessing cost of CoCo. Wall-clock time is reported in seconds and peak GPU memory in MB. This cost is incurred only once before federated training.

| Metric | Cora | Cite. | Pub. | Photo | Phy. | arxiv | Cham. | Squ. | CS | Tolo. | AR | Ques. | Reddit | OG-P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **PP-GSI Time (s)** | 3.72 | 4.96 | 3.07 | 4.32 | 9.41 | 2.41 | 3.39 | 2.86 | 5.76 | 2.62 | 4.96 | 5.75 | 3.75 | 31.36 |
| **GPU (MB)** | 37.55 | 86.65 | 19.55 | 24.66 | 186.97 | 12.02 | 44.27 | 32.75 | 83.21 | 12.07 | 20.89 | 42.93 | 32.21 | 325.63 |

Table 10: Node classification accuracy (%). Results are reported as mean ± standard deviation over five independent runs where available. The corresponding mean values are reported in Table 1. Δ denotes the difference between CoCo and the best-performing non-CoCo method.

| Method | Cora | Cite. | Pub. | Photo | Squ. | Tol. | AR |
|---|---|---|---|---|---|---|---|
| FedPub | 89.59 ± 0.33 | 70.74 ± 0.38 | 86.47 ± 0.85 | 92.76 ± 0.19 | 44.25 ± 0.51 | 78.84 ± 0.72 | 45.00 ± 3.61 |
| FedAvg | 88.37 ± 0.27 | 69.24 ± 0.70 | 85.55 ± 0.04 | 91.78 ± 0.22 | 39.98 ± 0.42 | 77.54 ± 0.20 | 45.04 ± 0.08 |
| FedGTA | 88.36 ± 0.30 | 69.15 ± 0.86 | 85.44 ± 0.06 | 91.74 ± 0.21 | 38.22 ± 0.16 | 77.92 ± 0.14 | 45.06 ± 0.19 |
| FedDC | 89.60 ± 0.49 | 75.17 ± 0.38 | 81.81 ± 0.32 | 91.56 ± 0.67 | 35.62 ± 0.74 | 78.21 ± 0.04 | 38.81 ± 0.27 |
| FedTGP | 87.38 ± 0.20 | 65.16 ± 0.52 | 84.48 ± 0.25 | 78.03 ± 3.92 | 42.85 ± 0.23 | 79.23 ± 0.04 | 49.08 ± 0.07 |
| FedTAD | 88.36 ± 0.31 | 69.33 ± 0.26 | 85.44 ± 0.06 | 91.88 ± 0.42 | 40.91 ± 0.40 | 78.26 ± 0.11 | 45.11 ± 0.31 |
| MOON | 88.51 ± 0.31 | 71.09 ± 0.51 | 85.64 ± 0.10 | 91.85 ± 0.32 | 40.44 ± 0.41 | 77.87 ± 0.47 | 44.54 ± 0.12 |
| Scaffold | 76.66 ± 0.77 | 75.26 ± 0.78 | 70.14 ± 0.64 | 84.89 ± 1.56 | 35.34 ± 0.50 | 77.54 ± 0.04 | 36.81 ± 0.65 |
| FedSage+ | 88.91 ± 0.41 | 74.50 ± 0.52 | 87.93 ± 0.18 | 90.01 ± 0.37 | 49.30 ± 0.63 | 78.81 ± 0.21 | 48.13 ± 0.44 |
| AdaFGL | 90.32 ± 0.36 | 78.40 ± 0.47 | 87.44 ± 0.22 | 89.49 ± 0.31 | 47.95 ± 0.58 | 78.47 ± 0.19 | 48.26 ± 0.39 |
| FedIIH | 82.66 ± 0.29 | 72.58 ± 0.43 | 86.58 ± 0.17 | 90.67 ± 0.34 | 28.35 ± 0.71 | 69.70 ± 0.27 | 50.24 ± 0.48 |
| CoCo | 92.04 ± 0.35 | 79.87 ± 0.19 | 88.10 ± 0.07 | 92.71 ± 0.68 | 49.80 ± 0.49 | 79.39 ± 0.04 | 52.21 ± 0.22 |
| Δ | +1.72 | +1.47 | +0.17 | −0.05 | +0.50 | +0.16 | +1.97 |

| Method | Ques. | CS | Phys. | Cham. | Arxiv | Reddit | OG-P |
|---|---|---|---|---|---|---|---|
| FedPub | 97.01 ± 0.00 | 89.40 ± 0.16 | 94.59 ± 0.07 | 58.22 ± 0.46 | 63.47 ± 0.45 | 92.96 ± 0.23 | 82.90 ± 0.38 |
| FedAvg | 97.01 ± 0.01 | 89.53 ± 0.07 | 94.31 ± 0.06 | 55.84 ± 0.38 | 61.35 ± 0.71 | 93.01 ± 0.20 | 82.75 ± 0.31 |
| FedGTA | 97.01 ± 0.00 | 89.46 ± 0.09 | 94.44 ± 0.01 | 55.13 ± 0.75 | 63.25 ± 0.86 | 94.10 ± 0.05 | 85.75 ± 0.27 |
| FedDC | 97.01 ± 0.00 | 90.25 ± 0.07 | 94.32 ± 0.10 | 49.81 ± 0.87 | 46.62 ± 0.23 | 91.46 ± 0.62 | 71.79 ± 0.56 |
| FedTGP | 97.04 ± 0.02 | 87.00 ± 0.14 | 93.89 ± 0.04 | 56.98 ± 0.19 | 59.07 ± 0.50 | 90.88 ± 0.22 | 80.54 ± 0.42 |
| FedTAD | 97.01 ± 0.00 | 89.32 ± 0.20 | 94.40 ± 0.08 | 55.84 ± 0.38 | 60.29 ± 0.18 | 92.97 ± 0.71 | OOM |
| MOON | 97.01 ± 0.00 | 89.91 ± 0.11 | 94.59 ± 0.04 | 56.55 ± 0.41 | 61.20 ± 0.04 | 92.85 ± 0.53 | 82.71 ± 0.35 |
| Scaffold | 97.01 ± 0.00 | 83.10 ± 0.06 | 93.64 ± 0.15 | 47.23 ± 1.01 | 37.25 ± 0.20 | 86.44 ± 0.07 | 61.85 ± 0.83 |
| FedSage+ | 97.03 ± 0.01 | 90.28 ± 0.13 | OOM | 60.53 ± 0.54 | 63.49 ± 0.37 | OOM | OOM |
| AdaFGL | 97.02 ± 0.01 | 91.61 ± 0.11 | 95.32 ± 0.08 | 56.68 ± 0.45 | OOM | OOM | OOM |
| FedIIH | 69.75 ± 0.34 | 91.61 ± 0.09 | 95.50 ± 0.06 | 42.01 ± 0.67 | OOM | OOM | OOM |
| CoCo | 97.08 ± 0.01 | 92.47 ± 0.10 | 95.95 ± 0.05 | 61.25 ± 1.46 | 64.70 ± 0.31 | 93.95 ± 0.28 | 85.91 ± 0.24 |
| Δ | +0.04 | +0.86 | +0.45 | +0.72 | +1.21 | −0.15 | +0.16 |

Table 9 separately reports the one-time cost of constructing the collaboration structure. Across most datasets, PP-GSI requires only a few seconds, while even the largest benchmark, ogbn-products, requires 31.36 seconds. Peak GPU memory remains modest, ranging from 12.02 MB to 325.63 MB across the evaluated datasets. Importantly, this cost is incurred only once before federated training and is therefore amortized over all subsequent communication rounds.

# B Additional Results

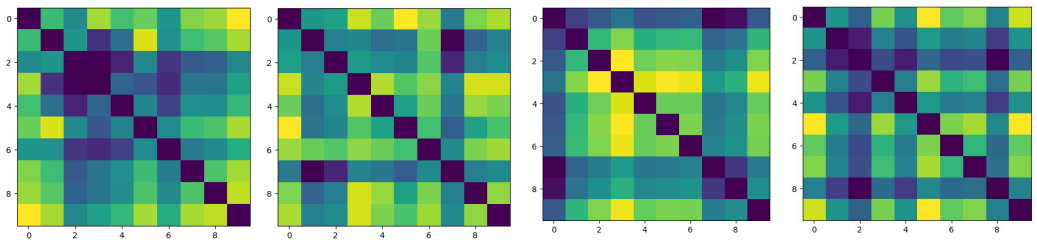
This section reports standard deviations for node classification results, supplementary visualizations, and ablations, including collaboration graphs, accuracy across communication rounds, client-count sensitivity, and coarsening-ratio effects.

## B.1 Standard Deviations for Node Classification Results

Table 10 reports the node classification results from Table 1 as mean ± standard deviation over five independent runs.

## B.2 Visualization of Collaboration Graph

We present here the visualization of the collaboration graph of the Citeseer, Cora, Chameleon and Squirrel datasets. Figure 7 illustrates the collaboration graph for these datasets in the form of a heat map.

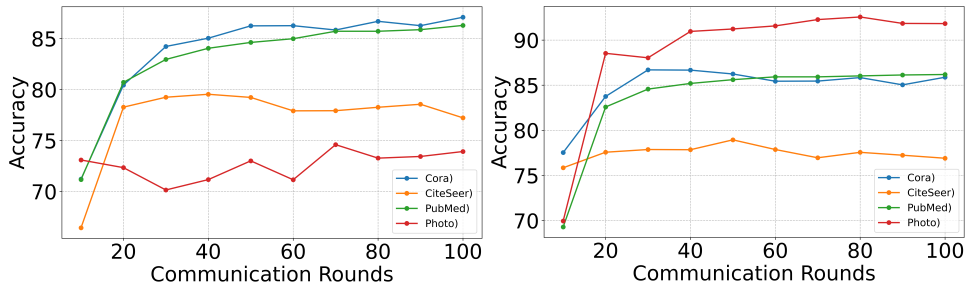


(a) Citeseer (b) Cora (c) Chameleon (d) Squirrel

Figure 7: Visualization of the inferred collaboration graphs. Each heatmap shows the interaction strength between pairs of clients, with stronger values indicating higher similarity and greater potential for collaboration. Results are shown here for 10 clients.

### B.3 Results for Varying Communication Rounds (Ablation).

Due to space constraints, we present here the Figure 8 which shows the convergence of CoCo with respect to communication rounds. Here the results are provided for the different GNN models, namely GCN and GAT (8 heads).




(a) (b)

Figure 8: Convergence of CoCo across communication rounds under different GNN backbones: (a) GCN and (b) GAT (8 heads).

### B.4 Effect of Varying Client Count on Baseline Methods

To provide a more comprehensive comparison, we extend our client-scaling study to include existing federated baselines FedPub, FedAvg, MOON, and Scaffold across three benchmark datasets: Cora, Chameleon, and Squirrel. Figure 9 presents the test accuracy trends as the number of participating clients increases from 5 to 25. Consistent with our observations for CoCo, all baselines experience a gradual performance decline as client count grows, primarily due to data fragmentation and reduced local data per client. Among the compared methods, FedPub and MOON maintain relatively higher stability, while Scaffold shows pronounced degradation under large federations.

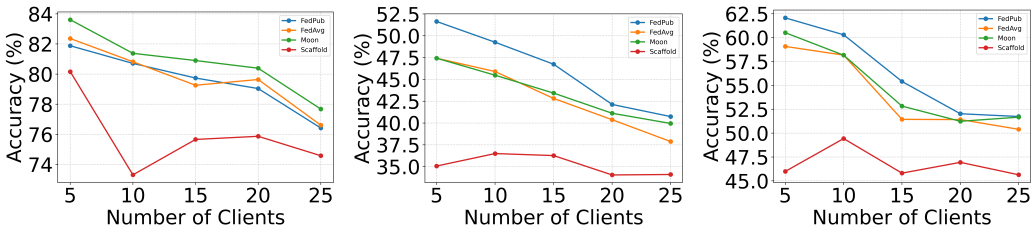
**Client Independence.** CoCo learns a *one-shot* collaboration graph before training, avoiding iterative inter-client synchronization. Hence, dropouts only skip aggregation in that round without changing the collaboration graph. Unlike pFL/FL methods that repeatedly update dependencies, CoCo is more robust to partial participation and better suited to asynchronous settings.

### B.5 Varying Coarsening Ratio for Scalable CoCo.

Scalability of CoCo with UGC coarsening.

### B.6 Anchor Generation Ablation.

We evaluate CoCo under different anchor generation schemes. Table 12 evaluates CoCo under different anchor constructions. *Syn-Gaussian* samples anchors i.i.d. as $a_k \sim \mathcal{N}(0, 1)$, while *SynOrtho-Gaussian* further orthogonalizes the sampled anchor directions to improve conditioning. *In-Data ($< d$)* draws in-distribution anchors from the dataset with #anchors $< d$. CoCo is effective across all choices, with In-Data performing best, indicating that in-distribution anchors are most informative for PP-GSI-based inference. Notably, CoCo remains strongly competitive with non-In-Data anchors that do not require client-private samples, supporting the use of public or governance-approved domain-compatible references




(a) (b) (c)

Figure 9: Test accuracy of baseline methods (FedPub, FedAvg, MOON, and Scaffold) with varying numbers of clients on (a) Cora, (b) Chameleon, and (c) Squirrel datasets.

Table 11: Node classification validation accuracy (%) under different coarsening ratios $\rho$.

| **Coar. Ratio ($\rho$)** | **Squirrel** | **Cora** | **CiteSeer** |
|---|---|---|---|
| 0.02 | 47.08 | 90.47 | 78.35 |
| 0.05 | 47.49 | 91.07 | 78.35 |
| 0.1 | 47.92 | 91.05 | 78.35 |
| 0.3 | 47.88 | 91.05 | 78.62 |
| 0.6 | 48.40 | 91.40 | 79.01 |
| 0.7 | 48.86 | 91.73 | 79.58 |
| 0.8 | 49.90 | 91.73 | 79.90 |
| 0.9 | 49.80 | 92.04 | 79.90 |
| 1.0 | 49.80 | 92.04 | 79.87 |

### B.7 Additional DP results.

We present the reconstruction behavior under node-level differential privacy (DP) in Fig. 10, which reports the Frobenius reconstruction error $\|X_{\text{frob}}\|$ as the number of anchors varies. For each anchor count, DP noise is added to the shared anchor-private point distance entries in the PP-GSI pipeline, and the reconstruction error is computed between the ground-truth private point embeddings $X_{priv}$ and the reconstructed $X_{priv}$ (Equation 2). The DP implementation and training details are provided in Appendix H. The results illustrate the interaction between two competing factors that determine reconstruction fidelity: the number of anchors and the magnitude of DP-induced perturbation in the incomplete distance matrix. Increasing the number of anchors provides additional geometric constraints, which improves reconstruction and drives the Frobenius error downward. This effect dominates in the low-anchor regime, where the error initially decreases as the anchor count increases. However, increasing the number of anchors also increases the number of shared DP noise injected distance entries. Beyond a certain point, the amount of DP noise perturbation begins to dominate the benefit of additional geometric constraints, causing the reconstruction error to increase again. Thus, the non-monotonic trend in Fig. 10 reflects a transition from an anchor-limited regime, where reconstruction improves with more anchors, to a noise-dominated regime, where additional noisy constraints degrade reconstruction. Importantly, the reconstruction error remains bounded away from zero under the evaluated DP mechanism, while the formal privacy guarantee is provided by the $(\epsilon_{\text{node}}, \delta_{\text{node}})$-LDP analysis. This indicates that node-level DP (CoCo-Specific 5.3) prevents exact feature recovery in the determined and overdetermined regimes as well. In the underdetermined regime, where $K < d$, privacy is supported by both geometric non-identifiability and DP perturbation, when $K \geq d$, the irrecoverability from underdetermination weakens, but DP noise continues to

Table 12: Anchor ablation for CoCo: test accuracy (%). Syn., Orth., and InD denote synthetic Gaussian, synthetic orthogonal Gaussian, and in-data anchors with $K < d$, respectively.

| **Dataset** | **Syn-Gaussian** | **SynOrtho-Gaussian** | **In-Data ($< d$)** |
|---|---|---|---|
| Cora | 92.01 | 92.04 | **92.04** |
| Pub. | 86.31 | 87.90 | **88.10** |
| Photo | 90.04 | 91.05 | **92.71** |

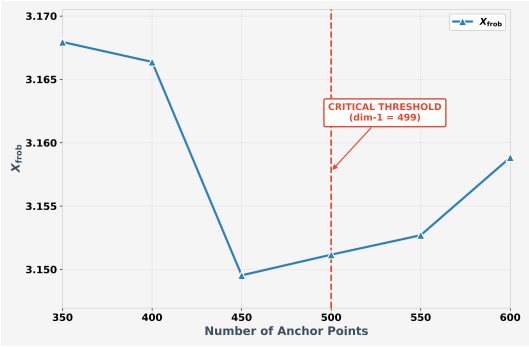



Figure 10: Frobenius reconstruction error $\|X\|_{\text{frob}}$ versus anchor count. The vertical marker at $K = d$ indicates the boundary beyond which the positive-dimensional non-identifiability guarantee established for $K < d$ no longer applies.

prevent exact reconstruction. Finally, although stronger perturbation reduces reconstruction fidelity, the downstream results in Table 4 show that the resulting degradation in node classification accuracy remains limited, indicating a favorable privacy-utility trade-off.

### B.8 Empirical Privacy Attack Evaluation

Geometric non-identifiability prevents unique reconstruction in the underdetermined regime ($K < d$), but does not rule out statistical inference or approximate reconstruction. We therefore evaluate CoCo's released anchor-distance representation on Cora and CiteSeer against membership inference Shokri et al. (2017), distribution/property inference Suri and Evans (2022), attribute inference Liu et al. (2022), approximate reconstruction, and link stealing He et al. (2021). The standard attack protocols are adapted to use the distance representation released by CoCo.

We compare the unperturbed representation (No DP) with node-level LDP at $\epsilon_{\text{node}} \in \{2, 0.1, 0.05, 0.01\}$, with $\delta_{\text{node}} = 10^{-7}$ and $\text{dp_seed} = 42$. The same sanitized representation is reused across all attacks for a given privacy setting. Since a separate sensitive attribute is unavailable, node class is used as the attribute-inference target.

**Attack setup.** Membership inference follows a black-box shadow-model attack Shokri et al. (2017); Liu et al. (2022). Property inference follows the white-box distribution-inference setting of Suri and Evans Suri and Evans (2022). Attribute inference uses an MLP attacker on the released representation. Approximate reconstruction uses multilateration with the public anchors, while link inference follows the supervised Attack-3 setting of He et al. He et al. (2021).

Table 13: ROC-AUC of membership, property, and link-inference attacks. Values near 0.5 indicate weak attack discrimination.

| Attack | Dataset | No DP | $\epsilon = 2$ | $\epsilon = 0.1$ | $\epsilon = 0.05$ | $\epsilon = 0.01$ |
|---|---|---|---|---|---|---|
| Membership | Cora | 0.652 | 0.501 | 0.512 | 0.499 | 0.500 |
| | CiteSeer | 0.571 | 0.521 | 0.543 | 0.555 | 0.537 |
| Property | Cora | 0.670 | 0.490 | 0.520 | 0.560 | 0.510 |
| | CiteSeer | 0.690 | 0.530 | 0.500 | 0.560 | 0.570 |
| Link | Cora | 0.752 | 0.569 | 0.568 | 0.567 | 0.569 |
| | CiteSeer | 0.820 | 0.559 | 0.557 | 0.559 | 0.559 |

**Inference attacks.** Table 13 shows that node-level LDP substantially reduces inference leakage. Membership-inference AUC drops from 0.652 to approximately 0.50 on Cora and from 0.571 to 0.52–0.56 on CiteSeer. Property-inference AUC similarly moves close to chance level on both datasets. Link inference is also substantially weakened, decreasing from 0.752 to approximately 0.57 on Cora and from 0.820 to approximately 0.56 on CiteSeer. The absence of a monotonic trend across $\epsilon_{\text{node}}$ is expected for empirical attacks, whose performance also depends on finite-sample training variability.

**Attribute inference and reconstruction.** LDP also substantially reduces attribute leakage: attribute-inference accuracy decreases from 58.20% to approximately 24% on Cora and from 54.61% to

Table 14: Attribute-inference accuracy (%) and mean cosine similarity of multilateration-based reconstruction. Lower attribute accuracy and cosine similarity near zero indicate weaker leakage.

| **Attack / Metric** | **Dataset** | **No DP** | $\epsilon = 2$ | $\epsilon = 0.1$ | $\epsilon = 0.05$ | $\epsilon = 0.01$ |
|---|---|---|---|---|---|---|
| Attribute Acc. (%) | Cora | 58.20 | 23.93 | 24.26 | 24.26 | 24.26 |
| | CiteSeer | 54.61 | 23.50 | 23.23 | 22.96 | 23.36 |
| Reconstruction Cos. | Cora | 0.6791 | -0.0014 | -0.0015 | -0.0015 | -0.0015 |
| | CiteSeer | 0.5950 | -0.0028 | -0.0028 | -0.0028 | -0.0028 |

approximately $23\%$ on CiteSeer. The effect on geometric reconstruction is particularly strong. Without DP, the mean reconstruction cosine similarity is $0.679$ on Cora and $0.595$ on CiteSeer, whereas under all evaluated privacy budgets its magnitude is approximately zero, indicating negligible directional similarity between the reconstructed and true features.

The small negative cosine values (e.g., $-0.0015$ and $-0.0028$) should not be interpreted as additional privacy beyond zero. Cosine similarity lies in $[-1, 1]$: negative values indicate opposite directional alignment, but values with such small magnitude are effectively indistinguishable from zero and therefore indicate no meaningful reconstruction alignment.

Overall, LDP substantially suppresses membership, property, attribute, reconstruction, and link leakage on both datasets. These empirical results complement the formal $(\epsilon_{\text{node}}, \delta_{\text{node}})$-LDP guarantee on the released anchor-distance representation.

### B.9 Model-level DP (OpenFGL).

We evaluate here CoCo under standard model-level DP using the OpenFGL framework Li et al. (2024c). While stronger privacy (smaller $\epsilon$) introduces additional noise, the resulting performance degradation remains limited, indicating a favorable privacy–utility trade-off. Implementation and training details in H.2.

Table 15: Performance under different privacy budgets.

| $\epsilon_{model}$ | **Pubmed** | | | | **Amazon-Ratings** | | | |
|---|---|---|---|---|---|---|---|---|
| | SCAFFOLD | MOON | FedAvg | CoCo | SCAFFOLD | MOON | FedAvg | CoCo |
| 1 | 69.43 | 71.33 | 71.56 | 71.86 | 36.84 | 37.67 | 37.71 | 37.71 |
| 5 | 70.29 | 71.33 | 71.57 | 71.97 | 36.92 | 37.69 | 37.75 | 37.84 |
| 10 | 70.30 | 71.36 | 72.64 | 72.64 | 36.95 | 37.69 | 37.79 | 37.92 |

## C Additional Metric and Implementation Details

We define evaluation metrics used for reconstruction quality and describes the partitioning schemes used to simulate heterogeneous federated graph settings.

### C.1 F-score and Frobenius Error.

We report two normalized metrics (Fig. 4a and 4b): (i)*F-score*, measuring neighborhood preservation between reconstructed and ground-truth private data point graphs:

$$\mathrm{FS} = \frac{2\,\mathrm{tp}}{2\,\mathrm{tp} + \mathrm{fp} + \mathrm{fn}},$$

where $\mathrm{tp}$, $\mathrm{fp}$, and $\mathrm{fn}$ are the counts of true positives, false positives, and false negatives, respectively (Egilmez et al., 2016), and (ii) *Frobenius reconstruction error*, quantifying embedding accuracy:

$$X_{\text{frob}} = \frac{\|X_{\text{priv}} - X'_{\text{priv}}\|_F}{\|X'_{\text{priv}}\|_F}$$

, where $X_{\text{priv}}$ and $X'_{\text{priv}}$ denote reconstructed and reference embeddings, respectively.

### C.2 Partition Schemes.

Specifically, we employ the following schemes:

- **Label skew:** Clients are assigned nodes with skewed class distributions, often generated using a Dirichlet prior to control concentration levels. This captures real-world imbalances in tasks such as fraud detection or recommendation.

- **Louvain community split:** Based on modularity optimization, Louvain detects dense communities within the graph, and these subgraphs are allocated to clients. This setting emphasizes topological heterogeneity, where clients correspond to structurally coherent but weakly interconnected clusters.
- **Metis community split:** A multilevel recursive bisection method that partitions the graph into communities with dense intra-client connectivity. Compared to Louvain, Metis allows tighter control of the number of partitions, aligning naturally with the number of clients.
- **Enhanced variants (Louvain+ and Metis+):** These strategies incorporate label distribution constraints during partitioning, thereby creating more challenging and realistic heterogeneous scenarios where both topology and labels jointly influence client splits.

# D THEOREM AND LEMMA PROOFS.

This section contains the proofs of Theorems 1, 2, and 3, Lemma 1, and Theorem 4.

## D.1 THEOREM 1 (STABILITY OF LEARNED GLOBAL STRUCTURE)

**Theorem** (Stability of Learned Global Structure). *Let $X_{\text{priv}}, \widehat{X}_{\text{priv}} \in \mathbb{R}^{N\times d}$ denote the original and reconstructed private-point embeddings, respectively, with $\|\widehat{X}_{\text{priv}} - X_{\text{priv}}\|_F \le \varepsilon_X$. Let $A^\star$ and $\widehat{A}$ denote the unique minimizers of Eq. 3 obtained using $X_{\text{priv}}$ and $\widehat{X}_{\text{priv}}$, respectively, and define $L_G^\star = \mathrm{Diag}(A^\star\mathbf{1}) - A^\star$. Assume that the rows of the two embedding matrices satisfy $\|x_i\|_2 \le R_X, \quad \|\widehat{x}_i\|_2 \le R_X, \quad i = 1,\dots,N$, and that the graph regularizer $f$ is $\mu$-strongly convex over $\mathcal{S}_A$. Then $\|\widehat{L}_G - L_G^\star\|_F \le c_G\varepsilon_X$, where $c_G = \frac{8R_X\sqrt{N(N+1)}}{\mu}$.*

*Proof.* Let $D(X) \in \mathbb{R}^{N\times N}, \quad D(X)_{ij} = \|x_i - x_j\|_2^2$, denote the matrix of squared pairwise Euclidean distances induced by the rows of an embedding matrix $X$. We bound the propagation of the reconstruction error through (i) the pairwise-distance matrix, (ii) the learned adjacency matrix, and (iii) the resulting graph Laplacian.

**Step 1: Stability of the pairwise-distance matrix.** Let $E = \widehat{X}_{\text{priv}} - X_{\text{priv}}$, and denote its $i$-th row by $e_i = \widehat{x}_i - x_i$. By assumption, $\|E\|_F = \|\widehat{X}_{\text{priv}} - X_{\text{priv}}\|_F \le \varepsilon_X$. For any $i, j$,

$$\left|D(\widehat{X}_{\text{priv}})_{ij} - D(X_{\text{priv}})_{ij}\right| = \left|\|\widehat{x}_i - \widehat{x}_j\|_2^2 - \|x_i - x_j\|_2^2\right| \le \|(\widehat{x}_i - \widehat{x}_j) - (x_i - x_j)\|_2 \\ (\|\widehat{x}_i - \widehat{x}_j\|_2 + \|x_i - x_j\|_2) = \|e_i - e_j\|_2 \left(\|\widehat{x}_i - \widehat{x}_j\|_2 + \|x_i - x_j\|_2\right).$$

Since $\|x_i\|_2 \le R_X, \quad \|\widehat{x}_i\|_2 \le R_X$ for every $i$, the triangle inequality gives

$$\|x_i - x_j\|_2 \le 2R_X, \qquad \|\widehat{x}_i - \widehat{x}_j\|_2 \le 2R_X.$$

Furthermore, $\|e_i - e_j\|_2 \le \|e_i\|_2 + \|e_j\|_2$. Therefore,

$$\left|D(\widehat{X}_{\text{priv}})_{ij} - D(X_{\text{priv}})_{ij}\right| \le 4R_X \left(\|e_i\|_2 + \|e_j\|_2\right).$$

Squaring both sides and using $(a+b)^2 \le 2(a^2+b^2)$ yields

$$\left|D(\widehat{X}_{\text{priv}})_{ij} - D(X_{\text{priv}})_{ij}\right|^2 \le 32R_X^2 \left(\|e_i\|_2^2 + \|e_j\|_2^2\right).$$

Summing over all $i, j$ gives

$$\|D(\widehat{X}_{\text{priv}}) - D(X_{\text{priv}})\|_F^2 \le 32R_X^2 \sum_{i,j=1}^{N} \left(\|e_i\|_2^2 + \|e_j\|_2^2\right) = 64NR_X^2 \sum_{i=1}^{N} \|e_i\|_2^2 = 64NR_X^2\|E\|_F^2.$$

Hence,

$$\|D(\widehat{X}_{\text{priv}}) - D(X_{\text{priv}})\|_F \le 8R_X\sqrt{N}\,\|\widehat{X}_{\text{priv}} - X_{\text{priv}}\|_F \le 8R_X\sqrt{N}\,\varepsilon_X. \tag{7}$$

**Step 2: Stability of the learned adjacency matrix.** For an arbitrary squared-distance matrix $D$, define

$$A(D) = \arg\min_{A \in \mathcal{S}_A} \{\langle D, A\rangle + f(A)\},$$

where $\langle D, A\rangle = \mathrm{tr}(D^\top A)$. By the definitions of $A^\star$ and $\widehat{A}$,

$$A^\star = A(D(X_{\mathrm{priv}})), \qquad \widehat{A} = A(D(\widehat{X}_{\mathrm{priv}})).$$

Define the extended-valued function $h(A) = f(A) + I_{\mathcal{S}_A}(A)$, where $I_{\mathcal{S}_A}$ is the indicator function of the convex feasible set $\mathcal{S}_A$. Since $f$ is $\mu$-strongly convex over $\mathcal{S}_A$, $h$ is $\mu$-strongly convex on its effective domain. The first-order optimality conditions for the two minimizers give

$$-D(\widehat{X}_{\mathrm{priv}}) \in \partial h(\widehat{A}), \qquad -D(X_{\mathrm{priv}}) \in \partial h(A^\star).$$

The subdifferential of a $\mu$-strongly convex function is $\mu$-strongly monotone. Therefore,

$$\left\langle -D(\widehat{X}_{\mathrm{priv}}) + D(X_{\mathrm{priv}}), \widehat{A} - A^\star \right\rangle \geq \mu \|\widehat{A} - A^\star\|_F^2.$$

By the Cauchy–Schwarz inequality,

$$\mu \|\widehat{A} - A^\star\|_F^2 \leq \|D(\widehat{X}_{\mathrm{priv}}) - D(X_{\mathrm{priv}})\|_F \, \|\widehat{A} - A^\star\|_F.$$

If $\widehat{A} = A^\star$, the desired bound is immediate. Otherwise, dividing by $\|\widehat{A} - A^\star\|_F$ gives

$$\|\widehat{A} - A^\star\|_F \leq \frac{1}{\mu} \|D(\widehat{X}_{\mathrm{priv}}) - D(X_{\mathrm{priv}})\|_F.$$

Using Eq. 7,

$$\|\widehat{A} - A^\star\|_F \leq \frac{8 R_X \sqrt{N}}{\mu} \varepsilon_X. \tag{8}$$

**Step 3: Stability of the learned graph Laplacian.** Recall that

$$\widehat{L}_G = \mathrm{Diag}(\widehat{A}\mathbf{1}) - \widehat{A}, \qquad L_G^\star = \mathrm{Diag}(A^\star \mathbf{1}) - A^\star.$$

Let $\Delta A = \widehat{A} - A^\star$. Then

$$\widehat{L}_G - L_G^\star = \mathrm{Diag}(\Delta A \mathbf{1}) - \Delta A. \tag{9}$$

Since both $\widehat{A}$ and $A^\star$ belong to $\mathcal{S}_A$, both have zero diagonal. Hence $\mathrm{diag}(\Delta A) = 0$. Therefore, $\mathrm{Diag}(\Delta A \mathbf{1})$ is supported only on the diagonal whereas $\Delta A$ is supported only off the diagonal. The two matrices are thus orthogonal with respect to the Frobenius inner product, and

$$\|\widehat{L}_G - L_G^\star\|_F^2 = \|\,\mathrm{Diag}(\Delta A \mathbf{1})\|_F^2 + \|\Delta A\|_F^2. \tag{10}$$

For every row $i$, the Cauchy-Schwarz inequality gives

$$\left(\sum_{j=1}^N \Delta A_{ij}\right)^2 \leq N \sum_{j=1}^N (\Delta A_{ij})^2.$$

Summing over $i$ yields $\|\,\mathrm{Diag}(\Delta A \mathbf{1})\|_F^2 = \|\Delta A \mathbf{1}\|_2^2 \leq N \|\Delta A\|_F^2$. Substituting into Eq. equation 10 gives

$$\|\widehat{L}_G - L_G^\star\|_F^2 \leq (N+1) \|\Delta A\|_F^2,$$

and therefore $\|\widehat{L}_G - L_G^\star\|_F \leq \sqrt{N+1}\, \|\widehat{A} - A^\star\|_F$. Finally, using Eq. equation 8,

$$\|\widehat{L}_G - L_G^\star\|_F \leq \sqrt{N+1} \frac{8 R_X \sqrt{N}}{\mu} \varepsilon_X = \frac{8 R_X \sqrt{N(N+1)}}{\mu} \varepsilon_X = c_G \varepsilon_X.$$

□

### D.2 THEOREM 2 (CLIENT-COARSENING STABILITY)

**Theorem** (Client-Coarsening Stability). *Let $\widehat{L}_G$ and $L_G^\star$ denote the recovered and true global operators, respectively, and let*

$$\widehat{L}_C = P\widehat{L}_G P^\top, \qquad L_C^\star = PL_G^\star P^\top,$$

*where $P = (S^\top S)^{-1}S^\top$ is the normalized client projection defined above. Let $n_{\min} = \min_{m=1,\dots,M} n_m$. Then*

$$\|\widehat{L}_C - L_C^\star\|_F \le \|P\|_2^2 \|\widehat{L}_G - L_G^\star\|_F = \frac{1}{n_{\min}}\|\widehat{L}_G - L_G^\star\|_F.$$

*Consequently, using Theorem 1,*

$$\|\widehat{L}_C - L_C^\star\|_F \le \frac{c_G}{n_{\min}}\varepsilon_X.$$

*Proof.* By definition,

$$\widehat{L}_C = P\widehat{L}_G P^\top, \qquad L_C^\star = PL_G^\star P^\top.$$

Therefore,

$$\widehat{L}_C - L_C^\star = P(\widehat{L}_G - L_G^\star)P^\top.$$

Using the standard operator–Frobenius norm inequality $\|ABC\|_F \le \|A\|_2\|B\|_F\|C\|_2$, we obtain

$$\|\widehat{L}_C - L_C^\star\|_F \le \|P\|_2^2 \|\widehat{L}_G - L_G^\star\|_F.$$

It remains to evaluate $\|P\|_2$. Let

$$N_C = S^\top S = \mathrm{Diag}(n_1, \dots, n_M).$$

Since

$$P = N_C^{-1}S^\top,$$

we have

$$PP^\top = N_C^{-1}S^\top S N_C^{-1} = N_C^{-1}.$$

Hence,

$$\|P\|_2^2 = \lambda_{\max}(PP^\top) = \lambda_{\max}(N_C^{-1}) = \frac{1}{n_{\min}},$$

where $n_{\min} = \min_{m=1,\dots,M} n_m$.

Therefore,

$$\|\widehat{L}_C - L_C^\star\|_F \le \frac{1}{n_{\min}}\|\widehat{L}_G - L_G^\star\|_F.$$

Applying Theorem 1,

$$\|\widehat{L}_G - L_G^\star\|_F \le C\epsilon_X,$$

gives

$$\boxed{\|\widehat{L}_C - L_C^\star\|_F \le \frac{c_G}{n_{\min}}\varepsilon_X}.$$

□

### D.3 Theorem 3 (Graph-Smoothing Property of CoCo Collaboration)

**Theorem** (Graph-Smoothing Property of CoCo Collaboration)**.** *Assume all clients have positive degree in $A_C$. For any locally updated parameters $\widetilde{\Theta} \in \mathbb{R}^{M\times p}$ and $\alpha \in [0,1]$, the CoCo aggregation*

$$\Theta^+ = [\alpha I + (1-\alpha)W]\widetilde{\Theta}$$

*satisfies*

$$\Theta^+ = \widetilde{\Theta} - (1-\alpha)\nabla\mathcal{E}_C(\widetilde{\Theta}), \qquad \mathcal{E}_C(\Theta^+) \le \mathcal{E}_C(\widetilde{\Theta}),$$

*where $\mathcal{L}_C = I - W$ and $\mathcal{E}_C(\Theta) = \frac{1}{2}\operatorname{tr}(\Theta^\top \mathcal{L}_C \Theta)$. Moreover, for $0 < \alpha < 1$, the inequality is strict whenever $\widetilde{\Theta}$ has a nonzero component outside $\ker(\mathcal{L}_C)$.*

*Proof.* Since $A_C$ is symmetric and nonnegative and $W = D_C^{-1/2} A_C D_C^{-1/2}$, the normalized collaboration Laplacian $\mathcal{L}_C = I - W$ is symmetric positive semidefinite, with eigenvalues $0 \le \lambda_k \le 2$.

Because $\mathcal{L}_C$ is symmetric,

$$\nabla\mathcal{E}_C(\Theta) = \mathcal{L}_C\Theta.$$

Using $W = I - \mathcal{L}_C$, the CoCo collaboration update can be rewritten as

$$\begin{aligned}\Theta^+ &= [\alpha I + (1-\alpha)W]\widetilde{\Theta} \\ &= [I - (1-\alpha)\mathcal{L}_C]\widetilde{\Theta} \\ &= \widetilde{\Theta} - (1-\alpha)\nabla\mathcal{E}_C(\widetilde{\Theta}).\end{aligned}$$

Thus, the implemented collaboration stage is exactly one gradient step on the normalized graph energy with step size $1-\alpha$.

To establish the smoothing property, let $\mathcal{L}_C = U\Lambda U^\top$, where $\Lambda = \operatorname{Diag}(\lambda_1, \dots, \lambda_M)$, and define $Y = U^\top\widetilde{\Theta}$. Then

$$\mathcal{E}_C(\widetilde{\Theta}) = \frac{1}{2}\sum_{k=1}^{M} \lambda_k \|Y_k\|_2^2.$$

After collaboration,

$$U^\top\Theta^+ = [I - (1-\alpha)\Lambda]Y,$$

and therefore

$$\mathcal{E}_C(\Theta^+) = \frac{1}{2}\sum_{k=1}^{M} \lambda_k\big(1-(1-\alpha)\lambda_k\big)^2\|Y_k\|_2^2.$$

For $\alpha \in [0,1]$ and $\lambda_k \in [0,2]$,

$$|1-(1-\alpha)\lambda_k| \le 1.$$

Hence every term in the graph energy is non-increasing, which gives

$$\mathcal{E}_C(\Theta^+) \le \mathcal{E}_C(\widetilde{\Theta}).$$

If $0 < \alpha < 1$ and $\lambda_k > 0$, then

$$|1-(1-\alpha)\lambda_k| < 1.$$

Thus, whenever $\widetilde{\Theta}$ contains a nonzero component associated with a positive eigenvalue of $\mathcal{L}_C$, equivalently a component outside $\ker(\mathcal{L}_C)$, the graph energy decreases strictly. □

### D.4 Lemma 1 (Non-Identifiability of Anchor-Distance Representations)

**Lemma** (Non-Identifiability of Anchor-Distance Representations, adapted from Malik et al. (2024))**.** *Let $X_{\text{priv}}$ denote the private data points and $X_A$ the set of $K$ anchors. If $K < d$, the original features $\{x_i\}$ cannot be uniquely reconstructed from the private point-anchor distances $\{\|x_i - a_j\|_2\}$. Hence, under these conditions, the anchor-distance representation does not uniquely determine the original private features.*

*Proof.* Consider a network in $d$-dimensional Euclidean space $\mathbb{R}^d$, comprising anchors $\mathcal{A} = \{a_1, \ldots, a_K\} \subset \mathbb{R}^d$ and non-anchor nodes $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$, with feature vectors $\boldsymbol{x_i} \in \mathbb{R}^d$. The locations of the anchors are known, while the locations of the non-anchors need to be estimated. Previous work Khan et al. (2009) shows that in $\mathbb{R}^d$, a minimum of $(d+1)$ anchors with known locations is required to locate $N$ non-anchor nodes. The use of anchors for distributed sensor localization is a well-researched area Khan et al. (2009). The study also assumed that all non-anchor points must be within the convex hull of the anchors. However, to accurately recover features in $\mathbb{R}^d$, at least $d$ anchors are necessary, even if non-anchors are placed in any location. Thus, having fewer than $d$ anchors, i.e., $K < d$, guarantees that exact feature embeddings cannot be obtained, ensuring privacy. □

### D.5 Theorem 4 (Node-Level Privacy of Distance-Based Representation)

**Theorem 4** (Node-Level Privacy of the Anchor-Distance Representation)**.** *Let*

$$\mathbf{d}(x) = [\|x - a_1\|_2, \ldots, \|x - a_K\|_2]^\top \in \mathbb{R}_+^K$$

*denote the complete anchor-distance representation of a private point $x \in \mathbb{R}^d$, where $\{a_1, \ldots, a_K\}$ are the reference anchors. Define the $\ell_2$-clipped representation*

$$\bar{\mathbf{d}}(x) = \mathbf{d}(x) \min\left\{1, \frac{C_{\text{vec}}}{\|\mathbf{d}(x)\|_2}\right\}, \tag{11}$$

*where $C_{\text{vec}} > 0$ is a fixed clipping radius.*
*Then the node-level $\ell_2$-sensitivity of the complete clipped anchor-distance representation satisfies*

$$\Delta_{2,\text{node}} := \sup_{x,x'} \left\|\bar{\mathbf{d}}(x) - \bar{\mathbf{d}}(x')\right\|_2 \leq \sqrt{2}\, C_{\text{vec}}. \tag{12}$$

*Consider the local randomization mechanism*

$$\mathcal{M}(x) = \bar{\mathbf{d}}(x) + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}\left(0, \tau_{\text{node}}^2 I_K\right), \tag{13}$$

*which is applied at the client before the representation is transmitted to the server, following the local-privacy model Duchi et al. (2018).*
*Let*

$$\bar{\Delta}_{2,\text{node}} = \sqrt{2}\, C_{\text{vec}},$$

*which is a valid upper bound on the global $\ell_2$-sensitivity from Eq. equation 12. If $\tau_{\text{node}} > 0$ is chosen such that*

$$\Phi\left(\frac{\bar{\Delta}_{2,\text{node}}}{2\tau_{\text{node}}} - \frac{\epsilon_{\text{node}}\tau_{\text{node}}}{\bar{\Delta}_{2,\text{node}}}\right) - e^{\epsilon_{\text{node}}}\Phi\left(-\frac{\bar{\Delta}_{2,\text{node}}}{2\tau_{\text{node}}} - \frac{\epsilon_{\text{node}}\tau_{\text{node}}}{\bar{\Delta}_{2,\text{node}}}\right) \leq \delta_{\text{node}}, \tag{14}$$

*where $\Phi$ denotes the cumulative distribution function of the standard normal distribution, then $\mathcal{M}$ satisfies $(\epsilon_{\text{node}}, \delta_{\text{node}})$-local differential privacy for the complete $K$-dimensional anchor-distance representation of one private point Balle and Wang (2018).*

*Proof.* We first establish the global $\ell_2$-sensitivity of the clipped anchor-distance representation. For any private point $x$, Eq. equation 11 implies

$$\left\|\bar{\mathbf{d}}(x)\right\|_2 \leq C_{\text{vec}}.$$

Furthermore, every coordinate of $\mathbf{d}(x)$ is a Euclidean distance and is therefore nonnegative. Since the clipping operation in Eq. equation 11 multiplies the complete vector by a nonnegative scalar, we also have

$$\bar{\mathbf{d}}(x) \in \mathbb{R}_+^K.$$

Consider any two possible private inputs $x$ and $x'$. Expanding their squared Euclidean difference gives

$$\begin{aligned}\left\|\bar{\mathbf{d}}(x) - \bar{\mathbf{d}}(x')\right\|_2^2 = \left\|\bar{\mathbf{d}}(x)\right\|_2^2 + \left\|\bar{\mathbf{d}}(x')\right\|_2^2 \\ - 2\bar{\mathbf{d}}(x)^\top \bar{\mathbf{d}}(x').\end{aligned} \tag{15}$$

Because both clipped distance vectors are element-wise nonnegative, their inner product is nonnegative:

$$\bar{\mathbf{d}}(x)^\top \bar{\mathbf{d}}(x') \geq 0.$$

Using this fact together with $\|\bar{\mathbf{d}}(x)\|_2 \leq C_{\text{vec}}$ and $\|\bar{\mathbf{d}}(x')\|_2 \leq C_{\text{vec}}$ in Eq. equation 15, we obtain

$$\begin{aligned}\left\|\bar{\mathbf{d}}(x) - \bar{\mathbf{d}}(x')\right\|_2^2 &\leq \left\|\bar{\mathbf{d}}(x)\right\|_2^2 + \left\|\bar{\mathbf{d}}(x')\right\|_2^2 \\ &\leq 2C_{\text{vec}}^2. \end{aligned} \tag{16}$$

Taking square roots yields

$$\left\|\bar{\mathbf{d}}(x) - \bar{\mathbf{d}}(x')\right\|_2 \leq \sqrt{2}\, C_{\text{vec}}.$$

Since the inequality holds for every pair $x, x'$, it follows that

$$\Delta_{2,\text{node}} = \sup_{x,x'} \left\|\bar{\mathbf{d}}(x) - \bar{\mathbf{d}}(x')\right\|_2 \leq \sqrt{2}\, C_{\text{vec}}.$$

We next establish the local differential-privacy guarantee. Define

$$f(x) = \bar{\mathbf{d}}(x).$$

The mechanism in Eq. equation 13 is therefore a Gaussian output-perturbation mechanism of the form

$$\mathcal{M}(x) = f(x) + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}\left(0, \tau_{\text{node}}^2 I_K\right).$$

For any two possible private inputs $x, x'$, let

$$D(x, x') = \|f(x) - f(x')\|_2.$$

From the sensitivity result above,

$$D(x, x') \leq \Delta_{2,\text{node}} \leq \bar{\Delta}_{2,\text{node}} = \sqrt{2}\, C_{\text{vec}}.$$

The analytic Gaussian mechanism of Balle and Wang Balle and Wang (2018) states that a Gaussian output-perturbation mechanism with global $\ell_2$-sensitivity $\Delta$ and noise standard deviation $\tau$ satisfies $(\epsilon, \delta)$-differential privacy if the Gaussian-CDF condition

$$\Phi\left(\frac{\Delta}{2\tau} - \frac{\epsilon\tau}{\Delta}\right) - e^{\epsilon}\Phi\left(-\frac{\Delta}{2\tau} - \frac{\epsilon\tau}{\Delta}\right) \leq \delta$$

holds.

Applying this result to $f(x) = \bar{\mathbf{d}}(x)$ and using the valid sensitivity upper bound

$$\bar{\Delta}_{2,\text{node}} = \sqrt{2}\, C_{\text{vec}},$$

Eq. equation 14 ensures that, for every pair of possible private inputs $x, x'$ and every measurable set $\mathcal{S} \subseteq \mathbb{R}^K$,

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon_{\text{node}}} \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta_{\text{node}}. \tag{17}$$

Because the randomization in Eq. equation 13 is performed locally at the client before the anchor-distance representation is transmitted to the server, Eq. equation 17 is precisely a node-level $(\epsilon_{\text{node}}, \delta_{\text{node}})$ local differential-privacy guarantee for the complete anchor-distance representation Duchi et al. (2018).

Therefore, $\mathcal{M}$ satisfies $(\epsilon_{\text{node}}, \delta_{\text{node}})$-LDP. □

## E Privacy Model and Federated Information Sharing

Federated learning does not require clients to reveal no information to the coordinating server, rather, its basic data-locality principle is that the underlying client records remain decentralized, while protocol-defined derived quantities may be communicated for collaborative learning. We follow this setting and assume an honest-but-curious coordinating server that follows the protocol and observes the quantities communicated to it. In CoCo, raw node features, labels, and local graph records remain at the clients, while clients communicate derived distance-based representations required to construct the collaboration structure.

This communication pattern is consistent with existing FL and personalized FL methods, which communicate different forms of derived information depending on the collaboration mechanism. Table 16 summarizes representative examples. The particular information communicated varies across methods, but in each case the federated boundary is defined by keeping the underlying client data local rather than by eliminating communication with the server.

Table 16: Representative information-sharing patterns in federated and personalized federated learning. Federated methods keep the underlying client records decentralized while communicating protocol-specific derived quantities needed for optimization, personalization, or collaboration.

| **Pipeline** | **Information communicated outside clients** | **Information retained locally** | **Role of communicated information** |
|---|---|---|---|
| FedAvg (McMahan et al., 2017a) | Model parameters or gradients | Raw training examples, features, and labels | Server-side model aggregation and global optimization |
| FedPub (Baek et al., 2023) | Local models and functional embeddings | Raw node features, labels, and local subgraphs | Estimating relationships among client models for personalized aggregation |
| pFedGraph (Ye et al., 2023) | Model similarities and client dataset sizes | Raw client training records | Constructing a client collaboration graph for personalized learning |
| FedGTA (Li et al., 2024b) | Topology-aware confidence and mixed-neighborhood statistics | Raw node features, labels, and local graph structure | Topology-aware personalized aggregation |
| GNN model sharing | Predictions, model parameters, or learned representations | Underlying node records and local graph data | Knowledge transfer or collaborative model learning |
| Clustered/ personalized FL (Sattler et al., 2020) | Model updates, embeddings, or client-similarity signals | Raw client datasets | Identifying related clients or client groups for personalized optimization |
| FedSage+ (Zhang et al., 2021) | Gradients, node embeddings, and generator outputs | Raw local node records and subgraphs | Sharing learned information for missing-neighbor generation and federated graph learning |
| Graph-private Sajadmanesh and Gatica-Perez, 2021 | Perturbed graph statistics or privately aggregated representations | Raw graph records | Collaborative graph learning through privacy-preserving derived information |
| CoCo | Distances from private points to anchors | Raw node features, labels, and local graph records | Inferring global structure and the client collaboration graph for personalized aggregation |

**Privacy scope.** Accordingly, the privacy boundary in CoCo distinguishes the private client records from the derived quantities intentionally released by the federated protocol. The coordinating server may use these released quantities and the resulting collaboration information for personalized aggregation, while the underlying node-level client data remain decentralized. The individual-level privacy guarantees associated with the released representations are developed in Section 4.

## F More Theoretical Guarantees with Proofs.

We provide additional privacy guarantees, including deterministic non-identifiability and information-theoretic leakage analysis.

**CoCo's Privacy Model.** We assume the server (and any eavesdropper observing the client-server channel) is *honest-but-curious*: it follows the protocol but may attempt to reconstruct private features or infer sensitive attributes from the observed distance messages and the resulting embeddings. Clients do *not* collude. This matches the standard FL threat model, except that clients transmit distance-based messages rather than gradient updates. We explicitly do *not* claim protection against colluding clients or active adversaries. We use the term "privacy-preserving" in a *geometric/information-theoretic* sense: (i) the released signals do not uniquely determine the underlying high-dimensional private

features, (ii) we can upper bound the *average* information revealed by the released signals under a probabilistic model.

**Notation.** In this section we analyze privacy for feature vectors in $\mathbb{R}^d$, where $d$ matches the feature dimension used in Algorithm 1. (If embeddings are used instead of raw features, the same analysis applies with $d$ replaced by the embedding dimension ($d$).)

For each non-anchor (NA) point $x \in \mathbb{R}^d$, the server does not observe $x$ directly. Instead, it only receives distances to a public anchor set $\mathcal{A} = \{a_1, \dots, a_K\} \subset \mathbb{R}^d$, i.e., the distance map below.

**Definition F.1.** *Let $\mathcal{A} = \{a_1, \dots, a_K\} \subset \mathbb{R}^d$ denote the public anchor set, and let $\Omega = \mathbb{R}^d \setminus \mathcal{A}$ denote the domain of non-anchor points. Define the anchor-distance map $\phi : \Omega \to \mathbb{R}^K, \quad \phi(x) = (\|x - a_1\|_2, \dots, \|x - a_K\|_2)^\top$. For $y \in \mathbb{R}^K$, the feasible set consistent with the released anchor distances is $\phi^{-1}(y) = \{x \in \Omega : \phi(x) = y\}$.*

**Assumption 1.** *We assume $K < d$ and the anchors are affinely independent, i.e., $\operatorname{rank}([\, a_2 - a_1,\ a_3 - a_1,\ \dots,\ a_K - a_1\,]^\top) = K - 1$. This is a standard non-degeneracy condition ensuring that $\phi$ has full row-rank Jacobian away from a measure-zero set (proved below).*

**Theorem 5** (Deterministic geometric non-identifiability)**.** *Let $\mathcal{A} = \{a_1, \dots, a_K\} \subset \mathbb{R}^d$ satisfy Assumption 1 and define $\phi$ as in Definition F.1. Then:*

1. *$\phi$ is smooth on $\Omega$, with Jacobian $D\phi(x) = \begin{bmatrix} \frac{(x-a_1)^\top}{\|x-a_1\|_2} \\ \vdots \\ \frac{(x-a_K)^\top}{\|x-a_K\|_2} \end{bmatrix} \in \mathbb{R}^{K \times d}$. Moreover, $D\phi(x)$ has full row rank $K$ for all $x \notin \operatorname{aff}(\mathcal{A})$.*
2. *For every regular value $y \in \mathbb{R}^K$ of $\phi$ (i.e., $\operatorname{rank}(D\phi(x)) = K$ for all $x \in \phi^{-1}(y)$), the level set $\phi^{-1}(y)$ is a smooth submanifold of $\mathbb{R}^d$ of dimension $\dim(\phi^{-1}(y)) = d - K$. In particular, since $K < d$, any nonempty $\phi^{-1}(y)$ contains uncountably many distinct points.*
3. *By Sard's theorem Sard (1942), the set of non-regular (critical) values has Lebesgue measure zero in $\mathbb{R}^K$. Hence, for almost every $y$ in the image of $\phi$, $\phi^{-1}(y)$ is $(d-K)$-dimensional and $\phi$ is non-injective.*

*Consequently, when $K < d$, exact recovery of $x$ from its anchor-distance vector $\phi(x)$ is generically impossible.*

*Proof.* Step 1. For each $k \in \{1, \dots, K\}$, $\phi_k(x) = \|x - a_k\|_2$. Since $x \in \Omega$, we have $x \neq a_k$ for every $k \in \{1, \dots, K\}$. Therefore, each component $\phi_k$ is smooth on $\Omega$, and hence $\phi$ is smooth on $\Omega$. Differentiating each component gives $\nabla_x \phi_k(x) = \frac{x - a_k}{\|x - a_k\|_2}$. Hence,

$$D\phi(x) = \begin{bmatrix} \frac{(x-a_1)^\top}{\|x-a_1\|_2} \\ \vdots \\ \frac{(x-a_K)^\top}{\|x-a_K\|_2} \end{bmatrix}.$$

Define

$$U(x) = \begin{bmatrix} (x-a_1)^\top \\ \vdots \\ (x-a_K)^\top \end{bmatrix},$$

and $R(x) = \operatorname{Diag}\left(\frac{1}{\|x-a_1\|_2}, \dots, \frac{1}{\|x-a_K\|_2}\right)$. Then $D\phi(x) = R(x)U(x)$. Since $x \in \Omega$, all $\|x - a_k\|_2 > 0$, and therefore $R(x)$ is invertible. Consequently, $\operatorname{rank}(D\phi(x)) = \operatorname{rank}(U(x))$. We now apply elementary row operations to $U(x)$ that preserve rank. Subtracting the first row from each of rows $2, \dots, K$ gives a rank-equivalent matrix with rows $(x-a_1)^\top, \quad (a_1-a_2)^\top, \dots, (a_1-a_K)^\top$. By Assumption 1, the $K-1$ vectors $\{a_1 - a_k\}_{k=2}^K$ are linearly independent. Therefore, $\operatorname{rank}(U(x)) = K$

whenever $x - a_1$ does not lie in their span, equivalently whenever $x \notin \mathrm{aff}(\mathcal{A})$. Hence, $\mathrm{rank}(D\phi(x)) = K$ for all $x \notin \mathrm{aff}(\mathcal{A})$.

Step 2. Let $y$ be a regular value of $\phi$. Since $\phi$ is smooth and $\mathrm{rank}(D\phi(x)) = K, \quad \forall x \in \phi^{-1}(y)$, the preimage theorem implies that $\phi^{-1}(y)$ is a smooth embedded submanifold of $\mathbb{R}^d$ of codimension $K$. Therefore, $\dim\big(\phi^{-1}(y)\big) = d - K$. Since $K < d$, we have $d - K \geq 1$. Hence, every nonempty regular level set $\phi^{-1}(y)$ contains uncountably many distinct points.

Step 3. By Sard's theorem, the set of critical values of the smooth map $\phi$ has Lebesgue measure zero in $\mathbb{R}^K$. Hence, for almost every feasible distance vector $y$ in the image of $\phi$, $y$ is a regular value. Therefore, $\dim\big(\phi^{-1}(y)\big) = d - K$. When $K < d$, this dimension is at least one, and thus the anchor-distance map is generically non-injective. Consequently, exact recovery of $x$ from its anchor-distance vector $\phi(x)$ is generically impossible when $K < d$. □

Under Assumption 1, if $K \leq d - 1$, then for almost every feasible distance vector $y$ the preimage $\phi^{-1}(y)$ is a smooth manifold of dimension at least 1, hence contains uncountably many candidates.

**Remark 2** (What happens at $K \geq d$)**.** *When $K \geq d$, the preimage is generically 0-dimensional (discrete). This does not imply unique recovery in general, but it reduces the deterministic ambiguity and may increase risk under auxiliary information, therefore we recommend $K \ll d$ in practice (Fig. 4a and 4b) while maintaining fidelity.*

**Remark 3.** *Theorem 5 formalizes deterministic geometric non-identifiability: when $K < d$, the anchor-distance vector $\phi(x)$ does not pin down a unique $x$, instead, there is generically a $(d - K)$-dimensional continuum of feature vectors indistinguishable to any adversary that only observes $\phi(x)$.*

We complement the deterministic ambiguity with an *average-case* leakage bound in a simplified probabilistic model. Locally linearizing $\phi$ around $x_0$ gives $\phi(x) \approx \phi(x_0) + D\phi(x_0)\,(x - x_0)$, so (up to an additive constant) the released signal behaves like a noisy linear observation with $A = D\phi(x_0) \in \mathbb{R}^{K \times d}$.

**Lemma 2** (Information-theoretic leakage under a linearized Gaussian model)**.** *Let $X \in \mathbb{R}^d$ be a random private feature vector and consider the linearized observation model*

$$Y = AX + N, \qquad X \sim \mathcal{N}(0, \Sigma_x), \quad N \sim \mathcal{N}(0, \Sigma_n), \quad N \perp X,$$

*where $\Sigma_x \succeq 0$ and $\Sigma_n \succ 0$. Then $(X, Y)$ is jointly Gaussian and $I(X, Y) = \frac{1}{2} \log\det\big(I + \Sigma_x A^\top \Sigma_n^{-1} A\big)$, Cover (1999) and Tse and Viswanath (2005). In the isotropic case $\Sigma_x = \sigma_x^2 I_d$ and $\Sigma_n = \sigma_n^2 I_K$, let $\lambda_1, \ldots, \lambda_r$ be the nonzero eigenvalues of $A^\top A$, where $r = \mathrm{rank}(A) \leq K$. Then $I(X, Y) = \frac{1}{2} \sum_{j=1}^{r} \log\Big(1 + \frac{\sigma_x^2}{\sigma_n^2} \lambda_j\Big)$. If $\|A\|_2^2 \leq c$ (equivalently, $\lambda_j \leq c$ for all $j$), then*

$$I(X, Y) \;\leq\; \frac{r}{2} \log\big(1 + c\,\mathrm{SNR}\big) \;\leq\; \frac{K}{2} \log\big(1 + c\,\mathrm{SNR}\big), \qquad \mathrm{SNR} := \frac{\sigma_x^2}{\sigma_n^2}.$$

*consequently, the per-coordinate leakage satisfies*

$$\frac{I(X, Y)}{d} \;\leq\; \frac{K}{2d} \log\big(1 + c\,\mathrm{SNR}\big),$$

*so for fixed* SNR *and bounded $\|A\|_2$ the leakage per coordinate vanishes as $K/d \to 0$.*

*Proof.* Since $X$ and $N$ are independent Gaussian, $(X, Y)$ is jointly Gaussian and $Y \sim \mathcal{N}\big(0,\, A\Sigma_x A^\top + \Sigma_n\big)$. For the linear Gaussian channel,

$$I(X, Y) = h(Y) - h(Y|X) = \frac{1}{2} \log \frac{\det(A\Sigma_x A^\top + \Sigma_n)}{\det(\Sigma_n)},$$

Cover (1999). Using the determinant identity

$$\det(A\Sigma_x A^\top + \Sigma_n) = \det(\Sigma_n)\, \det\big(I + \Sigma_x A^\top \Sigma_n^{-1} A\big),$$

we obtain

$$I(X,Y) = \frac{1}{2}\log\det\big(I + \Sigma_x A^\top \Sigma_n^{-1} A\big).$$

In the isotropic case, this becomes

$$I(X,Y) = \frac{1}{2}\log\det\Big(I + \tfrac{\sigma_x^2}{\sigma_n^2} A^\top A\Big).$$

Let $\lambda_1, \dots, \lambda_r$ be the nonzero eigenvalues of $A^\top A$ ($r = \operatorname{rank}(A)$). Then

$$\det\Big(I + \tfrac{\sigma_x^2}{\sigma_n^2} A^\top A\Big) = \prod_{j=1}^{r}\Big(1 + \tfrac{\sigma_x^2}{\sigma_n^2}\lambda_j\Big),$$

hence

$$I(X,Y) = \frac{1}{2}\sum_{j=1}^{r}\log\Big(1 + \tfrac{\sigma_x^2}{\sigma_n^2}\lambda_j\Big).$$

If $\|A\|_2^2 \le c$, then $\lambda_j \le c$ for all $j$, and therefore

$$I(X,Y) \le \frac{r}{2}\log\big(1 + c\,\mathrm{SNR}\big) \le \frac{K}{2}\log\big(1 + c\,\mathrm{SNR}\big).$$

Dividing by $d$ yields the per-coordinate bound.

Theorem 5 gives *deterministic ambiguity* (non-injectivity) when $K < d$, while Lemma 2 shows that under a linearized Gaussian model, the *average information leakage per coordinate* scales proportionally to $K/d$ (up to a $\log(1 + \mathrm{SNR})$ factor), supporting our use of $K \ll d$ in PP-GSI. □

## G Scaling CoCo for Large Graphs

As discussed in Section 3.4, the node-level PP-GSI formulation in Section 3.1 can become expensive for large graphs. To improve scalability, we first reduce each client's local graph to a smaller set of supernodes using Universal Graph Coarsening (UGC) Kataria et al. (2024), and then apply the same global-structure inference and client-level collaboration construction described in Sections 3.1 and 3.2 at the supernode level.

**Local Graph Coarsening.** Each client $C_m$ holds $X^{(m)} \in \mathbb{R}^{n_m \times d}$. We apply UGC locally to compress its $n_m$ nodes into $n'_m \ll n_m$ supernodes using randomized hashing and a max-occurrence rule. Concretely, UGC uses hash functions $h : \mathbb{R}^d \to \mathbb{Z}$,

$$h(x_i^{(m)}) = \left\lfloor \frac{1}{w}\left(r^\top x_i^{(m)} + b\right)\right\rfloor,$$

where $r \in \mathbb{R}^d$ is random, $b$ is a random offset, and $w > 0$ is the bin width. With multiple independent hashes, the resulting partition for client $C_m$ is denoted by

$$\mathcal{S}^{(m)} = \left\{S_1^{(m)}, \dots, S_{n'_m}^{(m)}\right\}, \qquad \bigcup_{u=1}^{n'_m} S_u^{(m)} = \{1, \dots, n_m\}.$$

Define the local coarsening matrix $\Pi^{(m)} \in \{0,1\}^{n'_m \times n_m}$ by

$$\Pi_{ui}^{(m)} = 1 \quad \Longleftrightarrow \quad i \in S_u^{(m)}.$$

The corresponding supernode representations are obtained by mean pooling:

$$z_u^{(m)} = \frac{1}{|S_u^{(m)}|}\sum_{i \in S_u^{(m)}} x_i^{(m)}, \qquad Z^{(m)} = D_{\Pi^{(m)}}^{-1}\Pi^{(m)}X^{(m)} \in \mathbb{R}^{n'_m \times d},$$

where

$$D_{\Pi^{(m)}} = \mathrm{Diag}\left(\Pi^{(m)}\mathbf{1}\right).$$

Let

$$N' = \sum_{m=1}^{M} n'_m, \qquad \Pi = \mathrm{blkdiag}\left(\Pi^{(1)}, \ldots, \Pi^{(M)}\right) \in \{0,1\}^{N' \times N}.$$

Stacking the local supernode representations gives

$$X_{SN} = \mathrm{blkcat}\left(Z^{(1)}, \ldots, Z^{(M)}\right) = D_{\Pi}^{-1} \Pi X_{\mathrm{priv}} \in \mathbb{R}^{N' \times d},$$

where

$$D_{\Pi} = \mathrm{Diag}(\Pi \mathbf{1}).$$

**Supernode-Level Global Structure Inference.** After local coarsening, we apply the same PP-GSI procedure from Section 3.1 to $\{X_{SN}, X_A\}$. Each client transmits the supernode-anchor distances

$$d(z_u^{(m)}, a_j) = \|z_u^{(m)} - a_j\|_2, \qquad 1 \le u \le n'_m, \quad 1 \le j \le K.$$

Distances among the anchors are directly available at the server. Using these observations, the server reconstructs $\widehat{X}_{SN} \in \mathbb{R}^{N' \times d}$ through the same anchored-SMACOF procedure described in Section 3.1. The global structure is then inferred over the reconstructed supernodes using the same graph-learning objective, yielding $\widehat{A}'$ and

$$\widehat{L}'_G = \mathrm{Diag}\left(\widehat{A}' \mathbf{1}\right) - \widehat{A}'.$$

Thus, the scalable formulation replaces the $N$ node-level representations used in Section 3.1 by $N' \ll N$ supernode representations, while leaving the global-structure inference procedure unchanged.

**Client-Level Collaboration from Supernodes.** To retain the same client-size normalization used in Section 3.2, let

$$S' \in \{0,1\}^{N' \times M}$$

denote the supernode-to-client membership matrix, where $S'_{um} = 1$ if supernode $u$ belongs to client $C_m$ and 0 otherwise. Since each supernode belongs to exactly one client,

$$S'^{\top} S' = \mathrm{Diag}\left(n'_1, \ldots, n'_M\right).$$

We therefore define the normalized client projection

$$P' = \left(S'^{\top} S'\right)^{-1} S'^{\top} \in \mathbb{R}^{M \times N'}.$$

Hence,

$$P'_{mu} = \begin{cases} 1/n'_m, & \text{if supernode } u \text{ belongs to } C_m, \\ 0, & \text{otherwise.} \end{cases}$$

Following the same construction as Section 3.2, the client-level coarsened operator is

$$\widehat{L}'_C = P' \widehat{L}'_G P'^{\top} \in \mathbb{R}^{M \times M}.$$

For two distinct clients $C_m$ and $C_\ell$,

$$(\widehat{L}'_C)_{m\ell} = -\frac{1}{n'_m n'_\ell} \sum_{u \in V'_m} \sum_{v \in V'_\ell} \widehat{A}'_{uv},$$

where $V'_m$ denotes the supernodes belonging to client $C_m$. Therefore, $-(\widehat{L}'_C)_{m\ell}$ represents the mean inferred affinity between the supernodes of clients $C_m$ and $C_\ell$, directly mirroring the node-level construction in Section 3.2.

The collaboration affinity is subsequently obtained using the same procedure as in the main formulation:

$$\bar{A}'_C = \mathrm{Diag}\left(\mathrm{diag}(\widehat{L}'_C)\right) - \widehat{L}'_C,$$

$$A'_C = \left[\frac{\bar{A}'_C + \left(\bar{A}'_C\right)^{\top}}{2}\right]_+.$$

Let

$$D'_C = \mathrm{Diag}(A'_C \mathbf{1}).$$

The collaboration matrix used for personalized aggregation is then

$$W = {D'_C}^{-1/2} A'_C {D'_C}^{-1/2},$$

which is used in the same personalized aggregation procedure as in the main CoCo formulation.

**Remark 4** (Node- vs. Supernode-Level Inference)**.** *The scalable formulation changes only the resolution at which global structure is inferred. When local coarsening is not required, we set*

$$\Pi^{(m)} = I, \qquad n'_m = n_m, \qquad N' = N.$$

*Consequently,*

$$S' = S, \qquad P' = P, \qquad \widehat{L}'_G = \widehat{L}_G, \qquad \widehat{L}'_C = \widehat{L}_C,$$

*and the scalable formulation reduces exactly to the node-level construction in Sections 3.1 and 3.2.*

# H Implementation Details for Node-Level Differential Privacy Experiments

This section provides the implementation details for the node-level local differential privacy mechanism used in the experiments of Section 5.3 and Appendix B.8.

## H.1 Approximate node-level Local Differential Privacy (LDP) for CoCo

For each private point $x_i$, CoCo constructs the complete $K$-dimensional private-to-anchor distance vector

$$\mathbf{d}_i = [d_{i1}, \ldots, d_{iK}]^\top, \qquad d_{ij} = \|x_i - a_j\|_2.$$

The privacy unit is one private point and its complete distance representation. Before transmission, the vector is $\ell_2$-clipped as

$$\bar{\mathbf{d}}_i = \mathbf{d}_i \min\left\{1, \frac{C_{\mathrm{vec}}}{\|\mathbf{d}_i\|_2}\right\}.$$

The clipping radius $C_{\mathrm{vec}}$ is determined only from the reference anchors, in our implementation, we use the 95th percentile of the $\ell_2$ norms of the anchor-to-anchor distance vectors. Thus, $C_{\mathrm{vec}}$ is fixed independently of the private points before any private release.

The clipped vector has node-level $\ell_2$ sensitivity bounded by

$$\bar{\Delta}_{2,\mathrm{node}} = \sqrt{2}\, C_{\mathrm{vec}}.$$

We then release

$$\widetilde{\mathbf{d}}_i = \bar{\mathbf{d}}_i + \boldsymbol{\eta}_i, \qquad \boldsymbol{\eta}_i \sim \mathcal{N}(0, \tau^2_{\mathrm{node}} I_K),$$

where $\tau_{\mathrm{node}}$ is numerically calibrated using the analytic Gaussian mechanism for the target $(\epsilon_{\mathrm{node}}, \delta_{\mathrm{node}})$ guarantee. Specifically, $\tau_{\mathrm{node}}$ is chosen such that

$$\Phi\left(\frac{\bar{\Delta}_{2,\mathrm{node}}}{2\tau_{\mathrm{node}}} - \frac{\epsilon_{\mathrm{node}}\tau_{\mathrm{node}}}{\bar{\Delta}_{2,\mathrm{node}}}\right) - e^{\epsilon_{\mathrm{node}}}\Phi\left(-\frac{\bar{\Delta}_{2,\mathrm{node}}}{2\tau_{\mathrm{node}}} - \frac{\epsilon_{\mathrm{node}}\tau_{\mathrm{node}}}{\bar{\Delta}_{2,\mathrm{node}}}\right) \le \delta_{\mathrm{node}}.$$

Negative perturbed distances are projected to zero before PP-GSI reconstruction. This projection is applied after privatization and is therefore post-processing of the LDP release.

**Privacy-budget evaluation.** We fix $\delta_{\mathrm{node}} = 10^{-6}$ and evaluate $\epsilon_{\mathrm{node}} \in \{16, 8, 4, 2, 0.1\}$ in the main node-classification experiment. Each setting reports the accuracy averaged over multiple seed runs.

## H.2 Model-Level DP (OpenFGL)

We use the OpenFGL framework Li et al. (2024c) to apply DP during federated training. DP is enforced via:

- gradient clipping,
- additive noise during local updates.

Specifically, each client clips gradients to a fixed norm and injects Laplace noise, ensuring privacy at the optimization level.

**Experimental setup.** We perform a sweep over: $\epsilon \in \{1, 5, 10\}$, with $\delta = 10^{-5}$, gradient clipping threshold = $1.0$. The noise scale follows the OpenFGL implementation:

$$b = \frac{\text{sensitivity}}{\epsilon/T},$$

where $T$ is the total number of training steps and sensitivity depends on the learning rate and clipping threshold.

**Training Configuration.** All experiments are conducted for $100$ communication rounds, across a federation of $10$ clients. We use the Adam optimizer with a learning rate of $0.01$ and a batch size of $128$. A fixed random seed of $42$ is used to ensure reproducibility.

**Implementation Details.** Model-level differential privacy is enabled during training by activating DP-specific configurations, including Laplace noise injection, gradient clipping, and privacy budget control. Specifically, we enable model DP training with Laplace noise, set the privacy budget $\epsilon$ accordingly, fix $\delta = 10^{-5}$, and use a gradient clipping threshold of $1.0$.

These two mechanisms operate at different stages of the pipeline and are evaluated independently to analyze robustness at both the representation and optimization levels.