Progressive Collaboration Pruning for Federated Learning

Siyao Cheng
SKLCCSE
Beihang University
Beijing, China
by2506402@buaa.edu.cn

Boyi Liu
SKLCCSE
Beihang University
Beijing, China
boyliu@buaa.edu.cn

Zimu Zhou
Department of Data Science
City University of Hong Kong
Hong Kong, China
zimuzhou@cityu.edu.hk

Zheng Wu
SKLCCSE
Beihang University
Beijing, China
zhengwu@buaa.edu.cn

Yongxin Tong
SKLCCSE
Beihang University
Beijing, China
yxtong@buaa.edu.cn

Abstract

Collaboration-based personalized federated learning (Co-PFL) mitigates data heterogeneity by enabling clients to selectively exchange knowledge through a learned collaboration graph. Despite their success, existing Co-PFL methods often update collaboration weights mainly from pairwise similarity, which can produce graphs that are either overly sparse and hinder knowledge propagation or overly dense and weaken personalization, leaving graph connectivity largely uncontrolled. In this paper, we propose FedPGP, a Co-PFL framework that explicitly schedules collaboration from dense to sparse over training rounds, enabling clients to first exploit broadly shared knowledge and then refine specialized knowledge aligned with their local distributions. FedPGP builds on a connectivity-aware similarity regularization that jointly accounts for global graph connectivity and client-model similarity, together with an efficient greedy pruning rule that removes edges with the least contribution to the regularized objective. We further present FedPGP+, a module-specific extension that learns different collaboration graphs for different model modules, allowing collaboration intensity to vary along model depth. Extensive experiments across diverse non-IID settings show that FedPGP and FedPGP+ consistently outperform state-of-the-art Co-PFL baselines, achieving up to 5.52% absolute accuracy improvement, while producing interpretable dense-to-sparse collaboration schedules.

CCS Concepts

• **Computing methodologies** → **Learning paradigms**.

Keywords

Federated learning, personalized federated learning, collaboration graph, graph pruning, non-IID data

ACM Reference Format:
Siyao Cheng, Boyi Liu, Zimu Zhou, Zheng Wu, and Yongxin Tong. 2026. Progressive Collaboration Pruning for Federated Learning. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 7–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799682.3840691

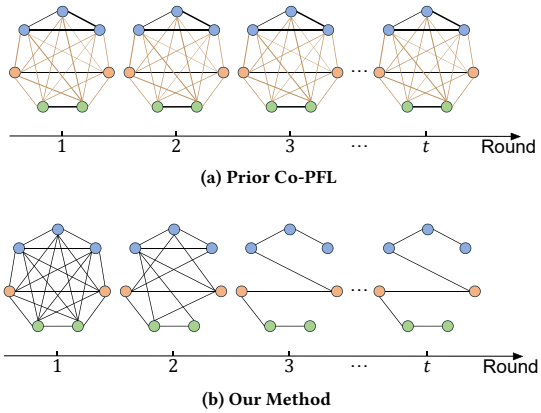


Figure 1: Conceptual comparison of existing Co-PFL and our proposed progressive collaboration pruning.

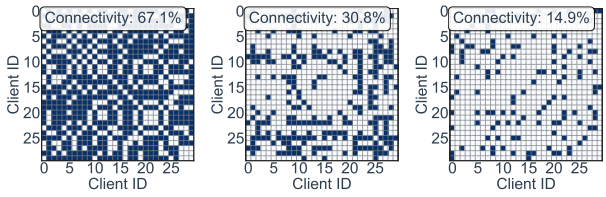
1 Introduction

Collaboration-based Personalized Federated Learning (Co-PFL) is an emerging paradigm for tackling non-IID client data by enabling *selective* knowledge sharing among clients [4, 24, 26, 43, 47]. Unlike conventional PFL that personalizes local objectives or model components [1, 3, 6, 30], Co-PFL personalizes *server-side aggregation* through a weighted *collaboration graph*. The graph is often inferred from model or parameter similarity [24, 26, 47], enabling each client to selectively aggregate knowledge from relevant peers.

Most Co-PFL methods [4, 24, 26, 43, 47] jointly train client models and update the collaboration graph, but they treat the *graph structure* as an *unconstrained* byproduct of training (see Fig. 1a). This design ignores a key property of non-IID data: distribution overlap across clients is often *layered*. Some overlap is broad and shared by many clients (*e.g.*, common features or prevalent classes), while other overlap is narrow and limited to a few clients (*e.g.*, rare classes or localized patterns). Dense collaboration can accelerate the learning of broad overlap through extensive signal averaging,

CIKM '26, Rome, Italy.

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840691



(a) Shallow (b) Middle (c) Deep

Figure 2: Collaboration graphs learned by FedPGP+ on CNN for CIFAR-10 under $Dir(0.1)$, showing pairwise connectivity in shallow, middle, and deep modules.

but it can also corrupt narrow overlap by injecting mismatched gradients, causing negative transfer on minority modes [38]. Therefore, beyond *who* to collaborate with, Co-PFL also faces a scheduling question: **how should collaboration evolve over training to balance shared representation learning and client-specific refinement?**

We advocate a **coarse-to-fine** collaboration schedule: prioritize broad sharing early, then progressively shift toward selective collaboration to mitigate interference and refine specialization. This perspective aligns with the learning dynamics in neural networks, where simpler patterns tend to be captured earlier than complex structures [25, 32, 39]. For example, spectral analyses indicates faster convergence on low-frequency components and curriculum-like behavior where easier patterns are learned before harder ones [34]. Translated to Co-PFL, these observations suggest that **early rounds should emphasize general knowledge transfer across many clients, while later rounds should reduce collaboration to stabilize client-specific adaptation.**

Guided by this principle, we propose FedPGP, **progressive collaboration pruning**, which explicitly schedules collaboration from dense to sparse (see Fig. 1b). We start from a fully-connected collaboration graph to encourage broad knowledge sharing. As training proceeds, we iteratively prune edges so that each client aggregates from a smaller set of increasingly relevant collaborators for specialization. While prior Co-PFL methods [4, 24, 26, 43, 47]can assign zero weights to irrelevant edges, they do not impose a *temporal* transition from generic dense collaboration to selective sparse collaboration. FedPGP enforces this transition via an explicit collaborator pruning mechanism guided by a regularizer that balances sharing and specialization over time.

We further develop FedPGP+, a **module-wise** extension that allows different model modules to maintain distinct collaboration graphs over the same client set. As shown in Fig. 2, the learned collaboration graph structure becomes increasingly sparse from shallow to deep modules: shallow modules retain dense graphs (favoring broadly shared representations), while deeper modules converge to selective and stable neighborhoods (favoring client-specific refinements). This *depth-dependent* pattern provides an interpretable support for our hypothesis that effective personalization requires collaboration to evolve from generic sharing toward specialization.

Our main contributions are as follows.

- We identify the collaboration scheduling gap in Co-PFL and motivate a coarse-to-fine principle based on layered data distribution overlap and neural learning dynamics.
- We propose FedPGP, a Co-PFL scheme that explicitly enforces dense-to-sparse collaboration. We further develop FedPGP+ with module-wise collaboration graphs, yielding interpretable collaboration patterns.
- Extensive evaluations show that we consistently outperform state-of-the-art Co-PFL methods [11, 14, 18, 20, 28, 35, 36, 43, 44, 47] by up to 5.52% in accuracy on both image classification and language understanding tasks.

2 Problem Statement

Consider collaboration-based personalized federated learning (Co-PFL) with n clients, where client i holds a private local dataset D_i. The datasets $\{D_1, \ldots, D_n\}$ are highly non-IID. The goal is to learn *personalized* models $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_n\}$ by enabling selective collaboration through a weighted graph $G(V, \boldsymbol{w})$, where V is the client set and $\boldsymbol{w} = [w_{ij}] \in \mathbb{R}^{n \times n}$ encodes pairwise collaboration strengths [4, 24, 26, 43, 47].

Formulation. Co-PFL often jointly optimizes the personalized models and collaboration weights:

$$\min_{\boldsymbol{\theta}, \boldsymbol{w}} \sum_{i=1}^{n} p_i \, \mathcal{L}\Big(\sum_{j=1}^{n} w_{ij}\theta_j; D_i\Big) + \mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w}) \tag{1}$$

where p_i is proportional to $|D_i|$, $\mathcal{L}(\cdot; D_i)$ is the empirical risk on client i, and $\sum_{j=1}^{n} w_{ij}$ is the personalized aggregation for client i. $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$ is a regularization that specifies how the collaboration relationship is estimated. It is often defined as the negation of certain similarity metric between client model parameters, *e.g.*, cosine similarity $\sum_{i,j=1}^{n} w_{ij} \cos(\theta_i, \theta_j)$ [47], and is either optimized with $\mathcal{L}(\cdot)$ *jointly* [13, 43, 47] or *iteratively* [11, 22, 36].

Workflow. We focus on *iterative* Co-PFL, where the server alternates between estimating collaboration weights and producing personalized models for the next round. Each communication round t consists of three phases: collaboration graph update, personalized model aggregation, and local training.

- **Collaboration Graph Update.** Given the client models $\{\theta_j^{t-1}\}_{j=1}^{n}$ at round t, the server updates the collaboration graph weights $\boldsymbol{w}$ by minimizing $\mathcal{R}_i(\cdot)$ in Eq. (1).

$$\begin{aligned} \min_{w_i} \quad & \mathcal{R}(\{\theta_j^{t-1}\}_{j=1}^{n}, \boldsymbol{w}) \\ \text{s.t.} \quad & \textstyle\sum_{j=1}^{n} w_{ij} = 1, \qquad w_{ij} \geq 0, \quad \forall j. \end{aligned} \tag{2}$$

- **Personalized Model Aggregation.** Based on the collaboration weights $\boldsymbol{w}$, the server aggregates client models from the previous round to produce personalized initialization $\widehat{\theta}_i^t$ to be dispatched to client i for the next round:

$$\widehat{\theta}_i^t = \sum_{j=1}^{n} w_{ij}\theta_j^{t-1} \tag{3}$$

- **Local Training.** Upon receiving the aggregated initialization $\widehat{\theta}_i^t$, each client performs local training on its private dataset D_i by minimizing the empirical risk:

$$\theta_i^t \leftarrow \widehat{\theta}_i^t - \eta \nabla F_i(\widehat{\theta}_i^t; D_i) \tag{4}$$

where $F_i(\cdot; D_i)$ is the local empirical loss and η is the learning rate. The updated models $\{\theta_i^t\}$ are then uploaded to the server for the next communication round.

Our Objective. While Eq. (2) estimates *who collaborates with whom* through $\boldsymbol{w}$, existing Co-PFL methods do not control the *density* of the collaboration graph. This can be suboptimal because overly *dense* collaboration may induce negative transfer by mixing incompatible clients, whereas overly *sparse* collaboration can isolate clients and slow down knowledge propagation [38]. Inspired by neural training dynamics [25, 32, 34, 39], we aim to design a Co-PFL algorithm that *progressively* controls collaboration density over rounds: *(i)* start with *broad* collaboration (a dense graph) to disseminate shared knowledge and stabilize early optimization; and *(ii)* gradually become *selective* by pruning unhelpful edges, so that each client ultimately collaborates with a small relevant neighborhood.

3 Method

3.1 FedPGP Overview

We propose FedPGP (**Fed**erated Learning with **P**rogressive-**G**raph **P**runing), a Co-PFL scheme that gradually transitions from broad sharing to selective cooperation.

Key Idea. Instead of estimating collaboration weights w_{ij} solely from *e.g.*, pairwise model similarity, FedPGP explicitly schedules the collaboration graph to evolve from *dense* to *sparse*. To enable direct control of graph connectivity, we parameterize the graph with per-client *collaboration sets* $\{C_1, \ldots, C_n\}$, where $C_i \subseteq \{1, \ldots, n\}$ denotes the clients whose models are aggregated to update client i. This design offers two benefits. *(i)* It provides an explicit handle to regulate each client's collaboration candidates (and thus the overall connectivity) across rounds. *(ii)* It converts collaboration-graph optimization from *continuous* weights w_{ij} to *discrete* sets C_i, allowing a simple greedy update rule. As we empirically show in Sec. 4.3, greedily updating C_i across rounds can effectively reduce the regularization $\mathcal{R}(\cdot)$, avoiding expensive per-round minimization (*e.g.*, quadratic programming) implied by Eq. (2).

New Components. FedPGP follows the three-phase workflow in Sec. 2, but replaces the collaboration-graph update with progressive pruning.

- We introduce a *connectivity-aware similarity regularization* $\mathcal{R}$ (Sec. 3.2) that jointly accounts for model similarity and graph connectivity when updating collaboration.
- Guided by $\mathcal{R}$, we progressively prune the graph at each round (Sec. 3.3) by removing, from each C_i, the edges that yield the smallest improvement measured by $\Delta\mathcal{R}$. This induces a monotonic dense-to-sparse transition during collaboration graph update. A detailed convergence analysis of our method is in Appendix A.2.
- To further examine dense-to-sparse scheduling, we propose a module-wise extension, FedPGP+, which allows different connectivity levels at different model depths (Sec. 3.4). We empirically show that (Sec. 4.2) the learned graphs tend to become sparse in deeper layers, and that this depth-varying connectivity improves accuracy over FedPGP, offering an interpretable validation of our scheduling hypothesis.

Algorithm 1: FedPGP

Input: Clients' model parameters $\theta_1, ..., \theta_n$, local dataset $D_1, ..., D_n$
Output: Personalized models $\theta_1^t, ..., \theta_n^t$

1 **for** *client* $i = 1, 2, \ldots n$ **do**
2 Initialize collaboration set $C_i \leftarrow \{1, \ldots, n\}$
3 **for** *round* $t = 1, 2, \ldots, T$ **do**
4 `// Local Training`
5 **for** *client* $i = 1, 2, \ldots, n$ **do**
6 $\theta_i^t \leftarrow \text{LocalTrain}(\widehat{\theta}_i^{t-1}, D_i)$
7 `// Progressive Pruning`
8 **for** *pruning counter* $c = 1, 2, \ldots, c_{max}$ **do**
9 Initialize candidate list $\mathcal{I} \leftarrow \emptyset$
10 **for** *each edge* (i, j) *with* $j \in C_i$ **do**
11 Compute $\Delta\mathcal{R}_{ij}$ using Eq. (9)
12 **if** $\Delta\mathcal{R}_{ij} < 0$ **then**
13 Add $(\Delta\mathcal{R}_{ij}, i, j)$ to candidate list $\mathcal{I}$
14 **if** $\mathcal{I}$ *is empty* **then**
15 **break**
16 Find $(i^*, j^*) \leftarrow \arg\min_{(i,j)} \Delta\mathcal{R}_{ij}$ in $\mathcal{I}$
17 Remove $C_i \leftarrow C_i \setminus \{j\}$
18 `// Collaboration Graph Update`
19 **for** *client* $i = 1, 2, \ldots, n$ **do**
20 $w_{ij} \leftarrow \frac{\mathbf{1}_{j \in C_i} |D_j|}{\sum_{j \in C_i} |D_j|}$
21 `// Personalized Aggregation`
22 **for** *client* $i = 1, 2, \ldots, n$ **do**
23 Aggregate $\widehat{\theta}_i^t$ using Eq. (3)

Workflow. Algorithm 1 summarizes FedPGP. We initialize each collaboration set as fully connected ($C_i^{(0)} = \{1, \ldots, n\}$), yielding a dense graph. In each round, clients perform local training and upload updated models to the server (Lines 3-6). The server then greedily prunes edges to improve $\mathcal{R}(\cdot)$ (Lines 8-17), reconstructs the collaboration weights (Lines 18-20), and performs client-wise personalized aggregation using the refined graph (Lines 21-23).

3.2 Connectivity-Aware Similarity Regularization

In FedPGP, we design a connectivity-aware regularization term for Eq. (1) that jointly accounts for *(i)* global graph connectivity and *(ii)* pairwise model similarity.

$$\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w}) = -(\alpha \cdot \Phi(\boldsymbol{w}) + (1 - \alpha) \cdot \boldsymbol{\theta}^\top \boldsymbol{w} \boldsymbol{\theta}) \tag{5}$$

where $\Phi(\boldsymbol{w})$ quantifies the connectivity of the collaboration graph, $\boldsymbol{\theta}^\top \boldsymbol{w} \boldsymbol{\theta}$ measures cross-client similarity, and $\alpha \in [0, 1]$ balances the two terms.

- **Connectivity term.** We measure connectivity using the *Fiedler value* [9, 29], *i.e.*, the second smallest eigenvalue of the graph Laplacian:

$$\Phi(\boldsymbol{w}) = \lambda_2(I - D^{-1/2} \boldsymbol{w} D^{-1/2}) \tag{6}$$

where D is the degree matrix of $\boldsymbol{w}$. A larger $\lambda_2(\cdot)$ indicates stronger global connectedness and thus more effective knowledge propagation, while $\lambda_2(\cdot) = 0$ implies that the graph is disconnected.

- **Similarity term.** The quadratic form can be written as $\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\theta_i^\top\theta_j$, which matches the similarity-driven collaboration objective in many Co-PFL methods [36, 47]. The inner product $\theta_i^\top\theta_j$ can be replaced by other similarity measures, such as cosine [13] or L_1/L_2 distances [26]. We empirically show that FedPGP remains effective under these alternatives (see Sec. 4.3).

Efficient Approximation of $\Delta\mathcal{R}$. As described in Sec. 3.3, our pruning rule requires evaluating the change $\Delta\mathcal{R}_{ij}$ by removing a candidate edge (i, j) in each round. For the similarity term, removing (i, j) yields

$$\Delta(\boldsymbol{\theta}^\top\boldsymbol{w}\boldsymbol{\theta}) \approx -2\, w_{ij}\, \theta_i^\top\theta_j \tag{7}$$

which can be computed directly.

Estimating the connectivity variation $\Delta\Phi(\boldsymbol{w})$ is more challenging, because it would require recomputing the algebraic connectivity of the Laplacian after each candidate removal. With $O(n^2)$ candidate edges, a naive implementation would incur $O(n^2)$ eigenvalue computations per round. To avoid this cost, we leverage the Fiedler vector $\boldsymbol{u}$ associated with λ_2. By first-order eigenvalue perturbation theory [9, 29], the connectivity change from removing (i, j) can be approximated as

$$\Delta\Phi(\boldsymbol{w}) \approx -\, w_{ij}\, (u_i - u_j)^2 \tag{8}$$

We provide a brief justification in Appendix A.1. Substituting this approximation into $\Delta\mathcal{R}_{ij}$ enables efficient pruning without repeated eigenvalue recomputation.

3.3 Collaboration Graph Pruning and Update

Instead of directly minimizing Eq. (2) over continuous weights w_{ij}, FedPGP adopts a greedy progressive pruning strategy. It iteratively removes edges from discrete collaboration sets $\{C_i\}$, and then deterministically reconstructs the corresponding weights from the updated sets.

Collaboration Graph Pruning. In each round, we assess how removing an edge (i, j) would change the regularization $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$. Let $\Delta\mathcal{R}_{ij}$ be the change in $\mathcal{R}$ after deleting (i, j):

$$\Delta\mathcal{R}_{ij} = -\alpha\Delta\Phi(\boldsymbol{w}) - (1-\alpha)\Delta(\boldsymbol{\theta}^\top\boldsymbol{w}\boldsymbol{\theta}) \tag{9}$$

We prune edges according to the following criteria:

- **Improving edges only.** We only consider edges whose removal reduces $\mathcal{R}$, i.e., $\Delta\mathcal{R}_{ij} < 0$.
- **Largest decrease first.** Among these candidates, we prioritize the edges with the most negative $\Delta\mathcal{R}_{ij}$ (*i.e.*, those whose removal yields the largest reduction in $\mathcal{R}$).

To control the sparsification rate, we prune at most $c_{\max}$ edges per round for each client as in Algorithm 1.

Graph Weight Update. Given the updated collaboration sets $\{C_i\}$, we reconstruct collaboration weights deterministically via normalized data-size weighting:

$$w_{ij} = \frac{\mathbf{1}_{j\in C_i}|D_j|}{\sum_{j\in C_i}|D_j|} \tag{10}$$

3.4 FedPGP+: Module-Specific Extension

Different modules of a deep model often encode features at different levels of generality: shallow layers capture broadly shared patterns, while deeper layers specialize to client-specific representations [2, 30]. Motivated by this observation, we extend FedPGP to FedPGP+, which allows collaboration to vary across model modules by learning *module-specific* collaboration graphs.

Module-Wise Collaboration Graphs. FedPGP+ partitions the model into a set of modules $\mathcal{M}$ and maintains an independent collaboration graph for each module $m \in \mathcal{M}$. For module m, we apply the same connectivity-aware regularization in Sec. 3.2:

$$\mathcal{R}(\boldsymbol{\theta}^m, \boldsymbol{w}^m) = -(\alpha\cdot\Phi(\boldsymbol{w}^m) + (1-\alpha)\cdot(\boldsymbol{\theta}^m)^\top\boldsymbol{w}^m\boldsymbol{\theta}^m) \tag{11}$$

where $\boldsymbol{w}^m$ is the collaboration graph for module m, and $\boldsymbol{\theta}^m$ collects the corresponding module parameters across clients. We then update each $\boldsymbol{w}^m$ using the same greedy pruning rule as in Sec. 3.3, but applied independently per module.

Module-Wise Personalized Aggregation. Given $\boldsymbol{w}^m$, the aggregated parameters of module m for client i at round t are

$$\widehat{\theta}_i^{m,t} = \sum_{j\in C_i} w_{ij}^m \theta_j^{m,t} \tag{12}$$

where C_i^m is the collaboration set induced by $\boldsymbol{w}^m$. The personalized model for client i is then obtained by assembling all module aggregates: $\widehat{\theta}_i^t = \mathrm{Concat}[\widehat{\theta}_i^{1,t}, \ldots, \widehat{\theta}_i^{|\mathcal{M}|,t}]$.

By allowing different modules to collaborate with different connectivity levels, FedPGP+ provides a flexible mechanism to combine broadly transferable features with client-specific specialization.

Process of FedPGP+. The process of FedPGP+ is detailed in Algorithm 2. The main difference of FedPGP+ is its module-wise collaboration mechanism, which includes:

- Module-specific collaboration set initialization (Lines 1-4),
- Independent progressive pruning for each module (Lines 9-11),
- Per-module graph weight updates (Line 12),
- Module-level personalized aggregation (Line 13),
- Final model reconstruction through module concatenation (Lines 14-16).

3.5 Analysis

Convergence Analysis. The convergence property of FedPGP is formally characterized in the following theorem, and the detailed proof is provided in Appendix A.2.

Theorem 1. *Under Assumptions A.1–A.4, FedPGP converges to a stationary point with the following convergence bound:*

$$\min_{t\in[T]}\sum_{i=1}^{n} p_i\mathbb{E}\left[\left\|\nabla F_i(\theta_i^{(t)})\right\|^2\right] \le O\left(\frac{1}{\sqrt{\tau T}}\right) + O\left(\frac{1}{T^{3/2}}\right) + O\left(\frac{\tau}{T}\right). \tag{13}$$

where n is the number of clients, T is the total number of communication rounds, τ is the number of local training steps, p_i is proportional to $|D_i|$, and the remaining constants are defined in Assumptions A.1–A.4.

Scalability. FedPGP scales to larger client populations since its extra cost is mainly incurred by server-side graph pruning. The graph update has a complexity of $O(n^2 \log n)$ for n clients and takes about 0.25s with 100 clients in our implementation. A complexity-based

Algorithm 2: FedPGP+

Input: Clients' module-wise parameters $\{\theta_1^m, \ldots, \theta_n^m\}_{m \in \mathcal{M}}$, local datasets $D_1, \ldots, D_n$
Output: Personalized models $\{\theta_1^t, \ldots, \theta_n^t\}$

```
// Initialization per module
for each module m ∈ M do
    for client i = 1, ..., n do
        Initialize collaboration set C_i^m ← {1, ..., n}
for round t = 1, 2, ..., T do
    // Local Training
    for client i = 1, ..., n do
        θ_i^t ← LocalTrain(θ̂_i^{t-1}, D_i)
    // Module-wise progressive pruning & aggregation
    for each module m ∈ M do
        Progressive pruning on C_i^m (as in Alg. 1)
        Update weights w_ij^m from C_i^m
        Aggregate θ̂_i^{m,t} using Eq. (12)
    // Merge modules
    for client i = 1, ..., n do
        θ̂_i^t ← Concat[θ̂_i^{1,t}, ..., θ̂_i^{|M|,t}]
```

estimate suggests that the latency remains only several seconds when scaling to 500 clients. In practice, the cost can be further reduced as the collaboration graph becomes progressively sparse during pruning.

Communication Overhead. FedPGP maintains the same communication pattern as standard iterative Co-PFL: each round clients upload their updated models and download personalized initializations. The transmitted message sizes are identical to vanilla Co-PFL methods, without introducing any additional communication overhead.

Privacy Preservation. Our method preserves the same privacy guarantees as standard federated learning. All client data remain on-device, and the server operates only on model parameters. The collaboration graph is constructed solely from uploaded model parameters, without accessing raw data or requiring additional privacy-sensitive information.

4 Experiment

4.1 Experiment Setup

Baselines. We compare with ten baselines: three generic FL methods (FedAvg [28], FedProx [20], MOON [18]), four clustered FL methods (CFL [36], IFCA [11], ICFL [44], CFL-GP [14]), and three Co-PFL methods (pFedGraph [47], FedSaC [43], FedSoft [35]).

For FedPGP, we set $\alpha = 0.5$ and $c_{\max} = 500$. For FedPGP+, we split the model into two modules (70%/30%).

Datasets and Models. We test three tasks. (i) Image classification: MNIST [16], FashionMNIST [42], EMNIST [5], and CIFAR-10 [15], using a 2-layer CNN. (ii) Language understanding: SST-2 [37] and IMDB [27], using a 12-layer BERT with hidden size 128. (iii) Object detection: BDD100K [48] and KITTI [10], using YOLOv5n. We simulate non-IID data via Dirichlet partitioning [12, 17] with $Dir(0.1)$ and $Dir(1.0)$. For classification and language tasks, we report test accuracy; for detection tasks, we report mAP@0.5.

Main results are reported on image classification, language understanding, and object detection tasks, while micro-benchmarks are conducted on image tasks. Unless otherwise specified, we report test accuracy.

Configurations. We simulate 100 clients with full participation (client sampling rate = 1). The number of communication rounds is $T = 200$, with $\tau = 5$ local epochs per round. We use a batch size of 64 and learning rate $\eta = 0.01$. SGD is adopted with momentum 0.9, weight decay 1×10^{-4}, and a learning rate decay of 0.99.

Environment. Experiments are run on an Intel Xeon Gold 6230R CPU and NVIDIA A100 GPUs with 40GB memory.

4.2 Main Results

Table 1, Table 2 and Table 3 report the performance of FedPGP and FedPGP+ on image classification, language understanding tasks, and object detection tasks, respectively. The best result is **bold-underlined** and the second-best is underlined.

Takeaways. We highlight four observations.

- **FedPGP achieves the best (or tied-best) accuracy across datasets and heterogeneity levels.** On the four image datasets, FedPGP improves the strongest baseline by up to 2.64% under $Dir(0.1)$ and 3.28% under $Dir(1.0)$. The trend also holds for language tasks, where FedPGP improves the strongest baseline by up to 1.35% (SST-2) and 0.50% (IMDB). On object detection, FedPGP improves the strongest baseline by up to 1.78% on BDD100K and 1.01% on KITTI in mAP@0.5. Overall, progressive collaboration pruning provides robust gains that generalize beyond a particular dataset type or modality.
- **FedPGP+ further improves accuracy via module-aware collaboration scheduling.** FedPGP+ further improves over FedPGP with consistent additional gains. The improvement reaches up to 3.55% under $Dir(0.1)$ and 3.84% under $Dir(1.0)$ on image tasks, and up to 1.50% on SST-2 and 1.20% on IMDB. On object detection, FedPGP+ further improves over FedPGP by up to 0.86% on BDD100K and 0.53% on KITTI. This suggests that *where* and *when* to specialize can differ across model components, making a single global schedule suboptimal.
- **Collaboration scheduling matters more as tasks become harder.** The gains are modest on simpler tasks (*e.g.*, on MNIST, FedPGP+ improves the strongest baseline by 0.46% under $Dir(0.1)$), but become notable on more complex datasets (up to 4.89% on CIFAR-10 and 2.75% on EMNIST under $Dir(0.1)$), where early broad collaboration helps learn transferable representations and later pruning enables specialization.
- **Collaboration scheduling outweighs complex collaboration metrics.** FedPGP outperforms FedSaC, which uses both similarity and complementarity signals, even though

Table 1: Accuracy comparison over image classification tasks.

Type	Methods	MNIST (%)		FashionMNIST (%)		CIFAR-10 (%)		EMNIST (%)	
		$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$
Generic FL	FedAvg	97.25±0.09	97.73±0.03	85.62±0.72	89.33±0.14	58.55±2.31	63.83±0.04	85.43±0.42	87.66±0.20
	FedProx	97.48±0.05	97.77±0.05	85.18±0.72	89.30±0.16	58.88±2.77	63.86±0.06	85.11±0.46	87.74±0.15
	MOON	97.44±0.15	97.73±0.07	83.92±1.27	89.14±0.28	58.20±2.81	62.95±0.11	83.99±0.65	87.82±0.15
Clustered FL	CFL	97.06±0.02	97.04±0.03	88.05±0.36	87.60±0.25	61.86±1.36	61.87±0.03	86.68±0.44	86.53±0.18
	IFCA	97.64±0.01	97.54±0.06	90.07±0.35	89.36±0.29	73.58±1.28	63.74±0.04	87.37±0.28	87.47±0.19
	ICFL	97.86±0.05	97.80±0.04	85.29±0.71	89.21±0.29	57.73±2.86	62.96±0.07	84.50±0.58	87.49±0.17
	CFL-GP	97.00±0.02	97.39±0.01	87.33±0.43	88.42±0.09	67.26±1.61	62.40±0.02	83.07±1.16	86.01±0.21
Co-PFL	pFedGraph	95.77±0.09	96.67±0.03	85.87±0.37	86.29±0.05	74.05±4.26	58.85±0.06	89.70±0.15	82.66±0.04
	FedSaC	96.07±0.42	93.90±0.74	89.91±0.35	86.60±0.64	77.61±1.05	52.34±0.43	91.02±0.44	87.27±0.63
	FedSoft	97.49±0.02	97.78±0.03	85.50±0.38	89.57±0.16	76.98±0.28	62.62±0.06	84.01±0.41	86.77±0.16
Ours	FedPGP	98.20±0.03	97.89±0.02	90.12±0.08	89.82±0.03	78.95±0.16	67.14±0.11	93.66±0.07	88.07±0.03
	FedPGP+	**98.32±0.05**	**98.22±0.04**	**92.70±0.22**	**89.85±0.08**	**82.50±0.38**	**67.52±0.40**	**93.77±0.07**	**91.91±0.14**

Table 2: Accuracy comparison over NLP tasks.

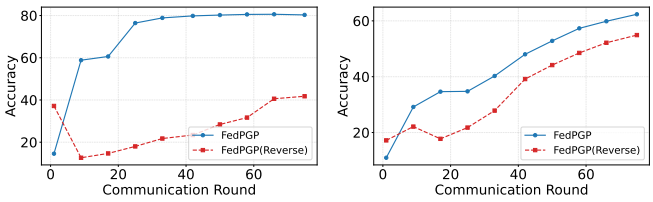
Type	Methods	SST-2 (%)		IMDB (%)	
		$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$
Generic FL	FedAvg	83.92±0.56	85.88±0.07	83.47±2.68	86.08±0.33
	FedProx	84.66±0.25	85.95±0.09	85.89±2.13	86.10±0.19
	MOON	75.10±0.38	85.10±0.18	74.49±2.06	85.92±0.21
Clustered FL	CFL	85.52±0.48	85.72±0.09	85.33±1.17	86.17±0.41
	IFCA	77.50±0.18	72.48±0.24	80.87±2.61	85.49±0.11
	ICFL	84.35±0.09	86.13±0.12	83.60±3.31	85.55±0.14
	CFL-GP	89.09±0.71	87.85±0.07	86.15±1.02	86.45±0.25
Co-PFL	pFedGraph	84.19±0.13	86.24±0.02	83.93±1.01	86.01±0.03
	FedSaC	88.75±0.89	87.45±0.33	86.06±1.31	85.45±0.06
	FedSoft	81.92±0.21	84.79±0.10	81.20±1.91	85.46±0.08
Ours	FedPGP	90.44±0.06	88.04±0.05	86.36±0.09	86.95±0.12
	FedPGP+	**91.94±0.03**	**88.15±0.05**	**87.56±0.04**	**87.22±0.09**

Table 3: mAP@0.5 comparison over object detection tasks.

Type	Methods	BDD100K (%)		KITTI (%)	
		$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$
Generic FL	FedAvg	26.16±1.33	25.42±0.42	56.67±0.47	66.39±0.11
	FedProx	25.96±2.12	25.53±0.18	58.27±0.38	66.13±0.07
	MOON	26.85±1.94	25.27±0.29	59.72±0.41	66.29±0.20
Clustered FL	CFL	26.61±0.87	24.76±0.37	58.02±0.51	65.94±0.12
	CFL-GP	25.13±1.58	24.11±0.16	60.92 ±0.44	66.50±0.10
	IFCA	21.33±3.62	24.73±0.19	56.72±0.59	65.27±0.29
	ICFL	26.23±1.17	25.36±0.28	58.23±0.14	66.30±0.08
Co-PFL	pFedGraph	26.26±0.74	25.27±0.43	57.89±0.15	66.46±0.04
	FedSaC	25.98±1.34	25.38±0.35	59.06±0.77	65.99±0.21
	FedSoft	25.83±0.58	25.21±0.73	59.53±0.31	66.58±0.05
Ours	FedPGP	27.91±0.03	27.31±0.08	61.93±0.09	67.38±0.07
	FedPGP+	**28.77±0.11**	**27.63±0.04**	**62.41±0.03**	**67.91±0.06**

our pruning criterion is similarity-based. A plausible explanation is that dense-to-sparse scheduling can implicitly induce complementarity over time.

- **Graph-based collaboration are more critical than clustering.** Clustered FL baselines that progressively split clients still lag behind our methods, likely due to knowledge isolation after cluster partitioning. For example, on CIFAR-10 under $Dir(0.1)$, FedPGP+ surpasses the best clustered method IFCA by 8.92%.



(a) CIFAR-10 $Dir(0.1)$ **(b) CIFAR-10** $Dir(1.0)$

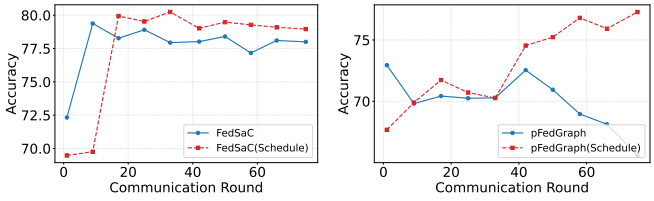
Figure 3: Impact of collaboration order.

4.3 Micro Benchmarks

Effectiveness of Dense-to-Sparse Collaboration. Fig. 3 examines whether the *order* of collaboration matters by reversing our progressive pruning. Instead of starting from a fully-connected graph and gradually removing edges, we initialize the graph as isolated and progressively *densify* it by adding edges according to model similarity.

The reversed schedule can yield slightly higher accuracy in the early rounds, as early selective collaboration induces faster short-term specialization. However, as training proceeds, it consistently underperforms FedPGP, and the gap persists even after many edges are added. These results suggest that early dense connectivity is important for learning transferable representations, whereas enforcing similarity-based collaboration too early leads to premature specialization and worse final accuracy.

Applicability of Collaboration Scheduling. We further test whether dense-to-sparse scheduling is useful by augmenting two strong Co-PFL baselines, pFedGraph and FedSaC, with our progressive scheduler while preserving their collaboration metrics.



(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 4: Applicability of collaboration scheduling.

Table 4: Impact of similarity metric.

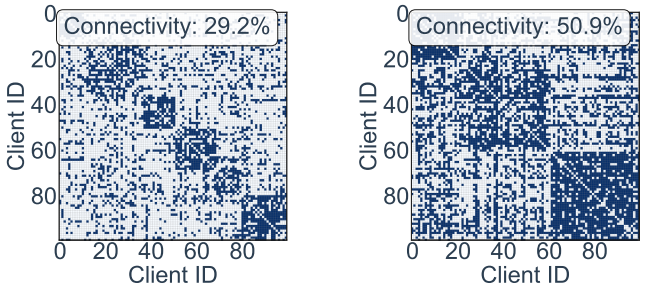
Similarity Metrics		CIFAR-10		EMNIST	
		$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$
Euclidean	Acc (%)	80.04	57.57	92.35	85.84
	Edge (%)	2.67	2.06	2.02	2.73
Cosine	Acc (%)	78.95	67.14	93.66	88.07
	Edge (%)	29.45	51.39	24.16	48.42
Pearson	Acc (%)	80.87	66.89	91.56	87.46
	Edge (%)	2.69	37.09	21.62	66.44

As shown in Fig. 4, the original methods converge slightly faster in early rounds but soon plateau. In contrast, their scheduled variants achieve higher final accuracy after an initial phase of enforced broad collaboration. This confirms that guiding collaboration from generic to specialized is a general strategy for boosting Co-PFL, orthogonal to the underlying collaboration metric.

Impact of Similarity Metric. Table 4 compares FedPGP under three client similarity metrics on CIFAR-10 and EMNIST. Overall, FedPGP achieves comparable accuracy across all three options. This suggests that progressive pruning is not tied to a particular similarity definition.

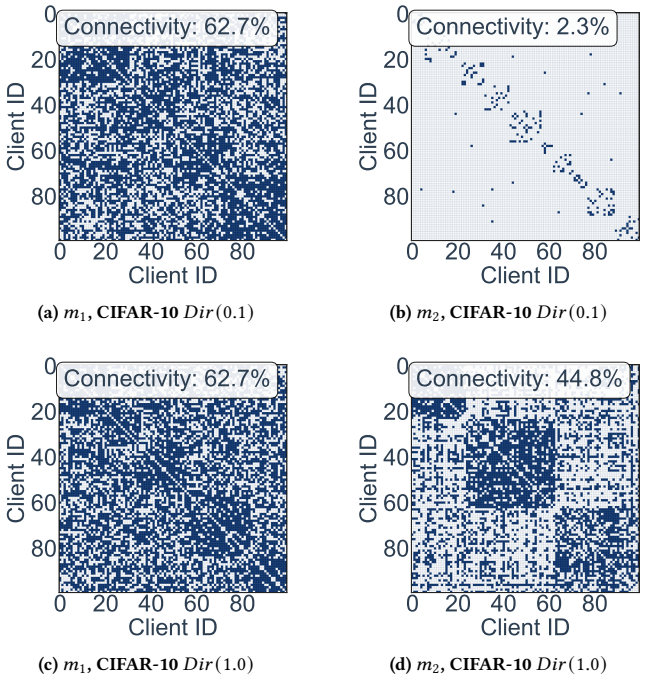
Collaboration Graph vs. Data Heterogeneity. Fig. 5 visualizes the learned collaboration graphs of FedPGP on CIFAR-10 under different non-IID settings. A higher connectivity ratio indicates stronger inter-client collaboration. The final graph is sparser under $Dir(0.1)$ (29.2%) and becomes denser under $Dir(1.0)$ (50.9%), suggesting that more aligned client data naturally promotes broader collaboration.

Collaboration Graph vs. Module Depth. Fig. 6 shows the learned collaboration graphs of FedPGP+ on CIFAR-10, highlighting its depth-adaptive collaboration. Under the highly non-IID setting ($Dir(0.1)$), m_1 remains dense (connectivity 62.7%), whereas m_2 is almost isolated (2.3%), indicating that FedPGP+ suppresses deep-layer collaboration to better preserve personalization. When the data distribution is more homogeneous ($Dir(1.0)$), m_1 stays similarly dense (62.7%) while m_2 becomes substantially more connected (44.8%), reflecting a shift to stronger cross-client sharing when client data are more aligned. Overall, FedPGP+ modulates collaboration across depth, maintaining broad sharing for general features while restricting deeper-layer collaboration in heterogeneous environments.



(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 5: Collaboration graphs learned by FedPGP.

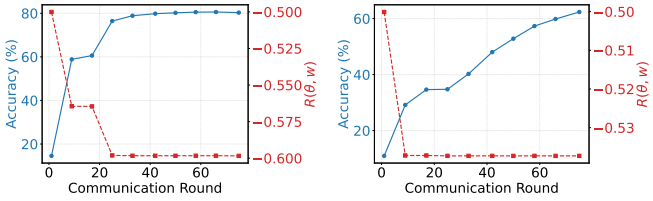


(a) m_1, CIFAR-10 $Dir(0.1)$ (b) m_2, CIFAR-10 $Dir(0.1)$

(c) m_1, CIFAR-10 $Dir(1.0)$ (d) m_2, CIFAR-10 $Dir(1.0)$

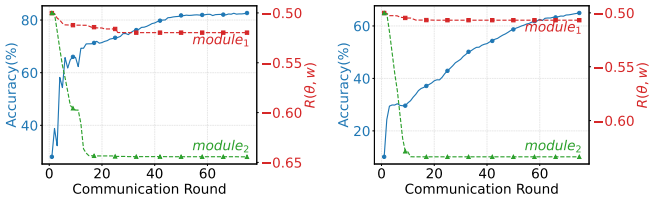
Figure 6: Collaboration graphs learned by FedPGP+.

Dynamics of Regularization Term. To verify that progressive pruning minimizes the regularization $\mathcal{R}$ and correlates with improved accuracy, we track $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$ and test accuracy during training. Fig. 7 and Fig. 8 plot their trajectories on CIFAR-10 for FedPGP and FedPGP+. In both cases, we observe a clear negative correlation: as $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$ decreases, accuracy increases. This suggests that minimizing $\mathcal{R}$ steers the system toward more coherent collaboration and better optimization. Moreover, $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$ converges more slowly under high heterogeneity ($Dir(0.1)$), since identifying stable collaboration relations requires more pruning. For FedPGP+, $\mathcal{R}$ differs across modules: the shallow module m_1 maintains higher values, consistent with stronger connectivity, whereas the deep module m_2 exhibits lower values, aligning with higher similarity and a stronger emphasis on local stability. Overall, these dynamics support the



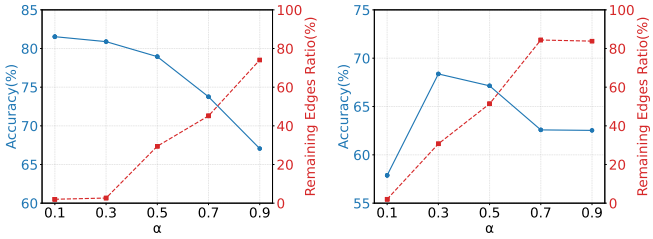
(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 7: Evolution of $\mathcal{R}(\theta, w)$ and accuracy of FedPGP.



(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 8: Evolution of $\mathcal{R}(\theta, w)$ and accuracy of FedPGP+.



(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 9: Impact of α on collaboration graph and accuracy.

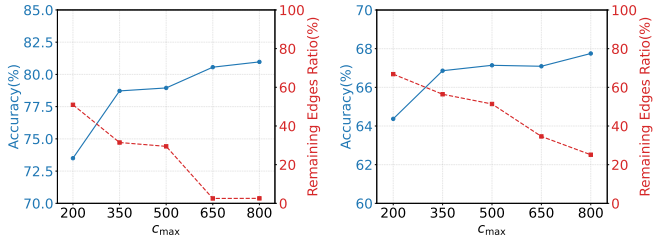
benefit of hierarchical scheduling in balancing global knowledge propagation and deep-layer personalization.

4.4 Impact of Hyperparameters

FedPGP has two hyperparameters: the balance weight α in Eq. (5) and the per-round pruning budget $c_{\max}$. FedPGP+ further introduces the module partition (number of modules m) and optionally module-specific weights α^m. We study how these choices affect collaboration graph and final accuracy.

Impact of α. Fig. 9 varies $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and reports accuracy and the remaining-edge ratio after pruning. Larger α yields denser graphs (stronger sharing), while smaller α promotes sparsity (stronger specialization).

Under high heterogeneity ($Dir(0.1)$), smaller α performs best, due to need for personalization. Under $Dir(1.0)$, accuracy peaks at a moderate α, indicating that some connectivity is still beneficial when heterogeneity is mild. Overall, $\alpha \in [0.3, 0.5]$ provides a good balance across settings.



(a) CIFAR-10 $Dir(0.1)$ (b) CIFAR-10 $Dir(1.0)$

Figure 10: Impact of $c_{\max}$ on collaboration graph and accuracy.

Table 5: Impact of Collaboration Item Number.

Methods	CIFAR-10 (%)		EMNIST (%)	
	$Dir(0.1)$	$Dir(1.0)$	$Dir(0.1)$	$Dir(1.0)$
FedPGP	78.95	67.14	93.66	88.07
FedPGP+ (2)	82.50 (+3.55)	67.52 (+0.38)	93.77(+0.11)	91.91 (+3.84)
FedPGP+ (3)	82.46 (+3.51)	67.60 (+0.46)	93.87 (+0.21)	91.29 (+3.22)
FedPGP+ (4)	83.22 (+4.27)	68.57 (+1.43)	94.41 (+0.75)	91.95 (+3.88)

Impact of $c_{\max}$. Fig. 10 varies $c_{\max} \in \{2, 4, 6, 8, 10\}$ and tracks accuracy and the convergence of the regularization $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{w})$. Larger $c_{\max}$ prunes more edges per round and thus produces sparser graphs. Accuracy improves with larger $c_{\max}$ under $Dir(0.1)$, but quickly saturates under $Dir(1.0)$. This suggests that aggressive pruning is more beneficial when heterogeneity is high, whereas mild heterogeneity only requires moderate pruning to retain effective knowledge sharing.

Impact of Module Number. Table 5 evaluates the partition granularity in FedPGP+ by using $m \in \{2, 3, 4\}$ modules (with ratios $\{70\%, 30\%\}$ for $m = 2$, $\{70\%, 15\%, 15\%\}$ for $m = 3$, and $\{70\%, 10\%, 10\%, 10\%\}$ for $m = 4$). Modular collaboration improves over FedPGP, reaching up to 3.55% gain on CIFAR-10 and 3.84% on EMNIST. The gains remain stable across different m, indicating that FedPGP+ is not sensitive to the exact partition granularity.

5 Related Work

Personalized Federated Learning. Federated learning enables multiple clients to collaboratively train models without directly sharing their local data [46]. In practice, federated systems exhibit substantial heterogeneity across clients, arising from differences in data distributions, local structures, and device capabilities, and even locally maintained representation spaces [33, 40, 45, 49]. Moreover, different clients or data providers may contribute unequally to collaborative learning, motivating efficient mechanisms for estimating their data utility [41]. Personalized Federated Learning (PFL) mainly addresses statistical heterogeneity by adapting the learned models to individual clients [17]. Existing PFL approaches follow two main directions. *(i) Model-based personalization* decomposes the model into shared and client-specific components, or designs personalized architectures to capture individual data characteristics [1, 6, 7, 19]. *(ii) Similarity-based personalization* aggregates

knowledge selectively according to inter-client similarity. Similar selective knowledge aggregation has also been explored in FedMosaic through query-adaptive adapter composition [21]. Among them, *clustering-based* PFL partitions clients into disjoint groups and shares models only within each cluster [8, 11, 14, 22, 36, 44]. *Collaboration-based* PFL (Co-PFL) learns a weighted graph that enables fine-grained pairwise aggregation [4, 24, 26, 43, 47]. Our work mainly focuses on the Co-PFL paradigm.

Collaboration-based Personalized Federated Learning. Compared with the hard partitioning in clustering-based PFL, Co-PFL advances personalization by enabling adaptive, pairwise knowledge fusion through a collaboration graph. This approach offers superior flexibility and facilitates more natural cross-client knowledge flow. Current Co-PFL methods typically construct this graph based on instantaneous client relationships, such as model parameter similarity [4, 23, 26, 31, 35, 47] or data distribution complementarity [43]. While this yields finer-grained collaboration than static clustering, it treats the graph structure as an unconstrained byproduct of each round's optimization. A key limitation of these approaches is the lack of explicit control over the evolution of graph density, which risks premature specialization and leads to suboptimal trade-offs between early-stage knowledge sharing and later-stage refinement. Our work addresses this by introducing progressive collaboration pruning, a scheduling mechanism that explicitly drives the collaboration graph from dense to sparse.

6 Conclusion

This paper revisits Co-PFL from a connectivity perspective and argues that effective collaborative personalization requires not only similarity-aware collaborator selection, but also explicit control of how collaboration connectivity evolves over training. We proposed FedPGP, a Co-PFL framework that schedules collaboration from dense to sparse, allowing clients to first benefit from broadly shared knowledge and then refine client-specific specialization. We further develop FedPGP+, a module-specific extension that learns different collaboration graphs across model modules, enabling collaboration intensity to vary along model depth. Extensive experiments on diverse non-IID settings show that FedPGP and FedPGP+ consistently outperform state-of-the-arts, and that the learned collaboration graphs provide an interpretable dense-to-sparse schedule that balances knowledge propagation and personalization. These results confirm explicit connectivity scheduling is essential for effective Co-PFL. Future work includes extending progressive collaboration scheduling to more heterogeneous model architectures, and exploring adaptive pruning policies that couple connectivity control with system constraints such as bandwidth and client availability.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (NSFC) (Grant Nos. 62425202, 62336003), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities No.JKF-2026007844235, and the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). The work is supported in part by a Theme-based Research Scheme (TRS) Project from the Research Grant Council in Hong Kong, under project T43-101/26-N. Yongxin Tong is the corresponding author.

A Appendix

A.1 Theoretical Analysis of $\Delta\Phi$ Approximation

We provide a justification for the approximation used to estimate the connectivity variation $\Delta\Phi(\boldsymbol{w})$ when removing an edge (i, j) from the collaboration graph. Let $L_{\boldsymbol{w}}$ denote the graph Laplacian associated with the symmetric adjacency matrix $\boldsymbol{w}$, and let λ_2 be its algebraic connectivity with corresponding Fiedler vector $\boldsymbol{u}$ with $\|\boldsymbol{u}\|_2 = 1$ and $\boldsymbol{u}^\top \mathbf{1} = 0$.

Remark 1 (Approximation of Connectivity Change). *Removing edge (i, j) with weight w_{ij} introduces a rank-one perturbation to the Laplacian:*

$$\Delta L = -w_{ij}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top, \tag{14}$$

where $\mathbf{e}_i$ is the i-th standard basis vector. According to first-order eigenvalue perturbation theory for symmetric matrices, the resulting change in λ_2 is approximated by

$$\Delta\lambda_2 \approx \boldsymbol{u}^\top(\Delta L)\,\boldsymbol{u} = -w_{ij}\,(u_i - u_j)^2. \tag{15}$$

Since the regularization term $\Phi(\boldsymbol{w})$ is defined to penalize reductions in connectivity, its variation obeys

$$\Delta\Phi(\boldsymbol{w}) \approx -\Delta\lambda_2 \approx w_{ij}\,(u_i - u_j)^2, \tag{16}$$

which provides an efficient surrogate for evaluating the structural importance of edge (i, j).

This approximation shows edges between clients with distinct Fiedler vector values dominate global connectivity, while nearby clients exert trivial impact. Incorporating this surrogate into pruning criteria eliminates repeated eigenvalue decompositions.

A.2 Theoretical Convergence Analysis

We denote the global objective function is defined as

$$F(\boldsymbol{\theta}^{(t)}) = \textstyle\sum_{i=1}^{n} p_i\, F_i(\theta_i^{(t)}), \tag{17}$$

where p_i is proportional to $|D_i|$ with $\sum_{i=1}^{n} p_i = 1$, and $F_i(\cdot)$ is the local empirical risk on client i's private dataset D_i. For notational simplicity, we define $\theta_i^{(t)} \equiv \theta_i^{(t,0)}$.

We then adopt four conventional assumptions from the FL theoretical literature [38].

Assumption A.1 (Smoothness). Each local loss function $F_i(\cdot)$ is L-smooth, i.e., for any θ, θ', there is $\|\nabla F_i(\theta) - \nabla F_i(\theta')\| \le L\|\theta - \theta'\|$.

Assumption A.2 (Bounded Below). $F_i(\cdot)$ is bounded below by F_{inf}, i.e., $F_i(\theta) \ge F_{\text{inf}}$ for all θ.

Assumption A.3 (Unbiased Gradient and Bounded Variance). For each client i, the stochastic gradient $g_i(\theta|\xi)$ is an unbiased estimate of $\nabla F_i(\theta)$ with bounded variance: $\mathbb{E}_\xi[g_i(\theta|\xi)] = \nabla F_i(\theta)$ and $\mathbb{E}_\xi[\|g_i(\theta|\xi) - \nabla F_i(\theta)\|^2] \le \sigma^2$.

Assumption A.4 (Collaboration-Induced Gradient Dissimilarity). There exist constants $\beta^2 \ge 1$ and $\kappa^2 \ge 0$ such that for any round t and collaboration weights $\{w_{ij}\}$:

$$\textstyle\sum_{i=1}^{n} \sum_{j \in C_i^{(t)}} w_{ij}^{(t)} \|\nabla F_i(\theta_i^{(t)}) - \nabla F_j(\theta_j^{(t)})\|^2$$
$$\le \beta^2 \textstyle\sum_{i=1}^{n} \|\nabla F_i(\theta_i^{(t)})\|^2 + \kappa^2.$$

Theorem A.5 (Optimization Bound of the Global Objective. Under Assumptions A.1–A.4, if the local learning rate η satisfies $\eta L \le \min\{\frac{1}{2\tau}, \frac{1}{\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\}$, then after T communication rounds, the expected average squared gradient norm of the aggregated personalized models is bounded by:

$$\begin{aligned}&\min_{t\in[T]} \textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] \\ &\le \tfrac{4[F(\boldsymbol{\theta}^{(0)})-F_{\inf}]}{\tau\eta T} + \tfrac{8L\eta\sigma^2}{T}\textstyle\sum_{t=0}^{T-1}\sum_{i=1}^n p_i \sum_{j\in C_i^{(t)}} (w_{ij}^{(t)})^2 \\ &\quad + \tfrac{6(\tau-1)\eta^2L^2\sigma^2}{\tau} + 12\tau(\tau-1)\eta^2L^2\kappa^2. \end{aligned} \tag{18}$$

Proof. We denote the mini-batch gradient by $g_k(\theta|\xi)$, and the full-batch gradient by $\nabla F_i(\theta)$. We further define the averaged mini-batch gradient for client i in round t as $\boldsymbol{d_i^{(t)}} = \frac{1}{\tau}\sum_{r=0}^{\tau-1} g_i(\theta_i^{(t,r)})$, and the averaged full-batch gradient as $\boldsymbol{h_i^{(t)}} = \frac{1}{\tau}\sum_{r=0}^{\tau-1} \nabla F_i(\theta_i^{(t,r)})$. The complete update from $\theta_i^{(t,0)}$ to $\theta_i^{(t+1,0)}$ can be expressed as:

$$\theta_i^{(t+1)} - \theta_i^{(t)} = \textstyle\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\theta_j^{(t)} - \theta_i^{(t)} - \tau\eta\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\boldsymbol{d}_j^{(t)}. \tag{19}$$

Main Proof. Using the L-smoothness of F, we have

$$\begin{aligned}&F_i(\theta_i^{(t+1)}) - F_i(\theta_i^{(t)}) \\ &\le \langle \nabla F_i(\theta_i^{(t)}), \textstyle\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\theta_j^{(t)} - \theta_i^{(t)}\rangle \qquad (20)\\ &\quad - \tau\eta\langle \nabla F_i(\theta_i^{(t)}), \textstyle\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\boldsymbol{d}_j^{(t)}\rangle \\ &\quad + \tfrac{L}{2}\|\textstyle\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\theta_j^{(t)} - \theta_i^{(t)} - \tau\eta\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\boldsymbol{d}_j^{(t)}\|^2. \qquad (21)\end{aligned}$$

Taking expectation and summing over all clients:

$$\begin{aligned}&\textstyle\sum_{i=1}^n p_i \mathbb{E}[F_i(\theta_i^{(t+1)}) - F_i(\theta_i^{(t)})] \\ &\le \underbrace{\textstyle\sum_{i=1}^n p_i \mathbb{E}[\langle \nabla F_i(\theta_i^{(t)}), \sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\theta_j^{(t)} - \theta_i^{(t)}\rangle]}_{T_1} \qquad (22)\\ &+ \underbrace{-\tau\eta\textstyle\sum_{i=1}^n p_i \mathbb{E}[\langle \nabla F_i(\theta_i^{(t)}), \sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\boldsymbol{d}_j^{(t)}\rangle]}_{T_2} \\ &+ \underbrace{\tfrac{L}{2}\textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\theta_j^{(t)} - \theta_i^{(t)} - \tau\eta\sum_{j\in C_i^{(t)}} w_{ij}^{(t)}\boldsymbol{d}_j^{(t)}\|^2]}_{T_3}. \qquad (23)\end{aligned}$$

Bounding T_1 in Eq. (23): Applying the polarization identity $2\langle a,b\rangle = \|a\|^2 + \|b\|^2 - \|a-b\|^2$:

$$\begin{aligned}T_1 &\le \tfrac{1}{2}\textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] \qquad (24)\\ &\quad + \tfrac{1}{2}\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\theta_j^{(t)} - \theta_i^{(t)}\|^2]. \qquad (25)\end{aligned}$$

Bounding T_2 in Eq. (23): Using the unbiasedness of stochastic gradients and the polarization identity, we have:

$$\begin{aligned}T_2 &= -\tfrac{\tau\eta}{2}\textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] \qquad (26)\\ &\quad - \tfrac{\tau\eta}{2}\textstyle\sum_{j=1}^n \mathbb{E}[\|\boldsymbol{h}_j^{(t)}\|^2] \sum_{i:j\in C_i^{(t)}} p_i w_{ij}^{(t)} \\ &\quad + \tfrac{\tau\eta}{2}\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\nabla F_i(\theta_i^{(t)}) - \boldsymbol{h}_j^{(t)}\|^2]. \qquad (27)\end{aligned}$$

Bounding T_3 in Eq. (23): Applying Jensen's inequality and Assumption A.3 gives:

$$\begin{aligned}T_3 &\le L\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\theta_j^{(t)} - \theta_i^{(t)}\|^2] \qquad (28)\\ &\quad + 2L\tau\eta^2\sigma^2 \textstyle\sum_{i=1}^n p_i \sum_{j\in C_i^{(t)}} {w_{ij}^{(t)}}^2 \\ &\quad + 2L\tau^2\eta^2 \textstyle\sum_{j=1}^n \mathbb{E}[\|\boldsymbol{h}_j^{(t)}\|^2] \sum_{i:j\in C_i^{(t)}} p_i w_{ij}^{(t)}. \qquad (29)\end{aligned}$$

Combining the bounds. We make the following simplification: Since the collaboration graph is typically symmetric or approximately symmetric, we assume $\sum_{i:j\in C_i^{(t)}} p_i w_{ij}^{(t)} \approx p_j$. This is a reasonable assumption in federated learning where collaboration is mutual. Under this assumption, we have:

$$\begin{aligned}&\textstyle\sum_{i=1}^n p_i \mathbb{E}[F_i(\theta_i^{(t+1)}) - F_i(\theta_i^{(t)})] \\ &\le (\tfrac{1}{2} - \tfrac{\tau\eta}{2})\textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] \qquad (30)\\ &+ (\tfrac{1}{2} + L)\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\theta_j^{(t)} - \theta_i^{(t)}\|^2] \\ &- \tfrac{\tau\eta}{2}(1 - 4L\tau\eta)\textstyle\sum_{j=1}^n p_j \mathbb{E}[\|\boldsymbol{h}_j^{(t)}\|^2] \qquad (31)\\ &+ 2L\tau\eta^2\sigma^2 \textstyle\sum_{i=1}^n p_i \sum_{j\in C_i^{(t)}} {w_{ij}^{(t)}}^2 \\ &+ \tfrac{\tau\eta}{2}\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \underbrace{\mathbb{E}[\|\nabla F_i(\theta_i^{(t)}) - \boldsymbol{h}_j^{(t)}\|^2]}_{T_4}. \qquad (32)\end{aligned}$$

Bounding the gradient difference term. We now focus on T_4. By the triangle inequality, we have:

$$\begin{aligned}&\tfrac{\tau\eta}{2}\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\nabla F_i(\theta_i^{(t)}) - \boldsymbol{h}_j^{(t)}\|^2] \\ &\le \tau\eta\textstyle\sum_{i=1}^n \sum_{j\in C_i^{(t)}} p_i w_{ij}^{(t)} \mathbb{E}[\|\nabla F_i(\theta_i^{(t)}) - \nabla F_j(\theta_j^{(t)})\|^2] \qquad (33)\\ &+ \tau\eta\textstyle\sum_{j=1}^n p_j \mathbb{E}[\|\nabla F_j(\theta_j^{(t)}) - \boldsymbol{h}_j^{(t)}\|^2]. \qquad (34)\end{aligned}$$

Following the standard local-SGD drift argument and using Assumptions A.1 and A.3, we obtain:

$$\mathbb{E}[\|\nabla F_j(\theta_j^{(t)}) - \boldsymbol{h}_j^{(t)}\|^2] \le \tfrac{L^2\tau\eta^2\sigma^2}{1-2L^2\tau^2\eta^2} + \tfrac{2L^2\tau^2\eta^2}{1-2L^2\tau^2\eta^2}\mathbb{E}[\|\nabla F_j(\theta_j^{(t)})\|^2]. \tag{35}$$

Final combination. Substituting Assumption A.4 and (35) into (32), and using the learning rate condition to ensure $1 - 4L\tau\eta \ge \frac{1}{2}$ and $\frac{2L^2\tau^2\eta^2}{1-2L^2\tau^2\eta^2} \le \frac{1}{2\beta^2}$, after detailed algebraic manipulation we obtain:

$$\begin{aligned}&\textstyle\sum_{i=1}^n p_i \mathbb{E}[F_i(\theta_i^{(t+1)}) - F_i(\theta_i^{(t)})] \\ &\le -\tfrac{\tau\eta}{4}\textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] + 2L\tau\eta^2\sigma^2 \sum_{i=1}^n p_i \sum_{j\in C_i^{(t)}} {w_{ij}^{(t)}}^2 \qquad (36)\\ &+ \tfrac{3}{2}(\tau-1)\eta^3L^2\sigma^2 + 3\tau(\tau-1)\eta^3L^2\kappa^2. \qquad (37)\end{aligned}$$

Summing over $t = 0,\dots,T-1$, applying Assumption A.2, and using $\min_t a_t \le \frac{1}{T}\sum_t a_t$ yields the result in Theorem A.5.

□

Corollary A.6 (Convergence of the Global Objective). By setting $\eta = \frac{1}{\sqrt{\tau T}}$, the bound in Eq. (18), we obtain

$$\min_{t\in[T]} \textstyle\sum_{i=1}^n p_i \mathbb{E}[\|\nabla F_i(\theta_i^{(t)})\|^2] = O(\frac{1}{\sqrt{\tau T}}) + O(\frac{1}{T^{3/2}}) + O(\frac{\tau}{T}).$$

As $T \to \infty$, the bound tends to zero, confirming convergence of the aggregated personalized models to stationary points.

B GenAI Usage Disclosure

The authors declare that no generative AI tools were used in any stage of this work, including research design, code implementation, data processing, experimental analysis, and manuscript writing.

References

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818* (2019).

[2] Yun-Hin Chan, Rui Zhou, Running Zhao, Zhihan JIANG, and Edith C. H. Ngai. 2024. Internal Cross-layer Gradients for Extending Homogeneity to Heterogeneity in Federated Learning. In *ICLR*.

[3] Hong-You Chen and Wei-Lun Chao. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *ICLR*.

[4] Jiayi Chen and Aidong Zhang. 2022. Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks. In *KDD*. 87–96.

[5] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *IJCNN*. IEEE, 2921–2926.

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting shared representations for personalized federated learning. In *ICML*. PMLR, 2089–2099.

[7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *NIPS* 33 (2020), 3557–3568.

[8] Kai Fang, Jiangtao Deng, Chengzu Dong, Usman Naseem, Tongcun Liu, Hailin Feng, and Wei Wang. 2025. MoCFL: Mobile Cluster Federated Learning Framework for Highly Dynamic Network. In *WWW*. 5065–5074.

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

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*. IEEE, 3354–3361.

[11] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *NIPS* 33 (2020), 19586–19597.

[12] Jonathan Huang. 2005. Maximum likelihood estimation of Dirichlet distribution parameters. *CMU Technique report* 76 (2005).

[13] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2021. Personalized cross-silo federated learning on non-iid data. In *AAAI*, Vol. 35. 7865–7873.

[14] Heasung Kim, Hyeji Kim, and Gustavo De Veciana. 2024. Clustered federated learning via gradient-based partitioning. In *ICML*.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 2002. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (2002), 2278–2324.

[17] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *ICDE*. IEEE, 965–978.

[18] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *CVPR*. 10713–10722.

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

[20] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *MLSys* 2 (2020), 429–450.

[21] Zhilin Liang, Yuxiang Wang, Zimu Zhou, Hainan Zhang, Boyi Liu, and Yongxin Tong. 2026. FedMosaic: Federated Retrieval-Augmented Generation via Parametric Adapters. In *SIGIR*. 1072–1082.

[22] Boyi Liu, Yiming Ma, Zimu Zhou, Yexuan Shi, Shuyuan Li, and Yongxin Tong. 2024. Casa: Clustered federated learning with asynchronous clients. In *KDD*. 1851–1862.

[23] Boyi Liu, Zimu Zhou, Cheng Fang, and Yongxin Tong. 2026. Federated Personalization of Early-Exit Networks. In *CIKM*.

[24] Boyi Liu, Zimu Zhou, Pengfei Gao, Shuo Kang, and Yongxin Tong. 2026. Towards Asynchronous Client Collaboration in Personalized Federated Learning. In *INFOCOM*.

[25] Ji Liu, Jiaxiang Ren, Ruoming Jin, Zijie Zhang, Yang Zhou, Patrick Valduriez, and Dejing Dou. 2024. Fisher Information-based Efficient Curriculum Federated Learning with Large Language Models. In *EMNLP*. 1–27.

[26] Jiahao Liu, Jiang Wu, Jinyu Chen, Miao Hu, Yipeng Zhou, and Di Wu. 2023. FedDWA: personalized federated learning with dynamic weight adjustment. In *IJCAI*. International Joint Conferences on Artificial Intelligence, 3993–4001.

[27] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*. 142–150.

[28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*. 1273–1282.

[29] Bojan Mohar, Y Alavi, G Chartrand, and Ortrud Oellermann. 1991. The Laplacian spectrum of graphs. *Graph theory, combinatorics, and applications* 2, 871-898 (1991), 12.

[30] Jae Hoon Oh, Sangmook Kim, and Seyoung Yun. 2022. FedBABU: Toward Enhanced Representation for Federated Image Classification. In *ICLR*.

[31] Yibo Pan, Boyi Liu, Zimu Zhou, Fan Gao, Tianyi Wang, Yi Xu, and Yongxin Tong. 2026. HiMAC: Hierarchical Federated Learning with Mobility-Aware Collaboration. In *CIKM*.

[32] Zhuang Qi, Yuqing Wang, Zitan Chen, Ran Wang, Xiangxu Meng, and Lei Meng. 2022. Clustering-based curriculum construction for sample-balanced federated learning. In *CICAI*. Springer, 155–166.

[33] Lehao Qu, Shuyuan Li, Zimu Zhou, Boyi Liu, Yi Xu, and Yongxin Tong. 2025. DarkDistill: Difficulty-Aligned Federated Early-Exit Network Training on Heterogeneous Devices. In *KDD*. 2374–2385.

[34] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *ICML*. PMLR, 5301–5310.

[35] Yichen Ruan and Carlee Joe-Wong. 2022. Fedsoft: Soft clustered federated learning with proximal local updating. In *AAAI*, Vol. 36. 8124–8131.

[36] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems* 32, 8 (2020), 3710–3722.

[37] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*. 1631–1642.

[38] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *NIPS* 33 (2020), 7611–7623.

[39] Mingjie Wang, Jianxiong Guo, and Weijia Jia. 2023. Federated Multi-Phase Curriculum learning to synchronously correlate user heterogeneity. *IEEE Transactions on Artificial Intelligence* 5, 5 (2023), 2026–2039.

[40] Yuxiang Wang, Yongxin Tong, Zimu Zhou, Ziyuan He, Ruixi Hu, and Ke Xu. 2026. Federated Retrieval over Embedding-Heterogeneous Vector Databases. In *ICDE*. IEEE, 350–363.

[41] Shuyue Wei, Yongxin Tong, Zimu Zhou, Tianran He, and Yi Xu. 2025. Efficient data valuation approximation in federated learning: A sampling-based approach. In *ICDE*. IEEE, 2922–2934.

[42] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[43] Kunda Yan, Sen Cui, Abudukelimu Wuerkaixi, Jingfeng Zhang, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. 2024. Balancing similarity and complementarity for federated learning. *ICML* (2024).

[44] Yihan Yan, Xiaojun Tong, and Shen Wang. 2023. Clustered federated learning in heterogeneous environment. *IEEE Transactions on Neural Networks and Learning Systems* 35, 9 (2023), 12796–12809.

[45] Linghua Yang, Wantong Chen, Xiaoxi He, Shuyue Wei, Yi Xu, Zimu Zhou, and Yongxin Tong. 2024. FedGTP: exploiting inter-client spatial dependency in federated graph-based traffic prediction. In *KDD*. 6105–6116.

[46] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 1–19.

[47] Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. 2023. Personalized federated learning with inferred collaboration graphs. In *ICML*. PMLR, 39801–39817.

[48] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*. 2636–2645.

[49] Tianlong Zhang, Xiaoxi He, Yuxiang Wang, Yi Xu, Rendi Wu, Zhifei Wang, and Yongxin Tong. 2025. FedMetro: Efficient metro passenger flow prediction via federated graph learning. In *KDD*. 5215–5224.