Rethinking Personalized Federated Learning in LLM Era: A Knowledge Sharing *

Boyi Liu

City University of Hong Kong & Beihang University

Abstract

Personalized Federated Learning (PFL) addresses the fundamental challenge of statistical data heterogeneity in federated learning by enabling clients to obtain customized models while benefiting from collaborative training. In this survey, we organize the diverse landscape of PFL methods around a three-dimensional knowledge-sharing taxonomy: (1) *What to Share* — the form of knowledge exchanged, ranging from partial parameters and class prototypes to knowledge distillation outputs; (2) *Who to Share With* — the selection of collaboration partners through hard clustering, soft clustering, or dynamic grouping; (3) *How to Share* — the aggregation mechanism, including similarity-weighted fusion, graph-based propagation, and hypernetwork generation. We systematically review representative methods under this framework, covering both traditional model-based FL and recent federated large language model (LLM) fine-tuning via LoRA adapters. We further discuss benchmark datasets, evaluation protocols, open challenges, and future research directions.

1 Introduction

Federated learning (FL) [McMahan *et al.*, 2017] enables multiple clients to collaboratively train a model while keeping raw data local, offering a principled solution to data privacy constraints in distributed environments. The arrival of the large language model (LLM) era has dramatically widened the scope of FL: the community is now asked not only to train modest classification models in a federated fashion, but to fine-tune billion-parameter foundation models [Zhang *et al.*, 2023e] across heterogeneous clients with highly diverse tasks. This shift invites us to *revisit* the field of personalized federated learning (PFL) with fresh eyes — asking which design principles established over the past decade carry over, and which need rethinking.

A unifying challenge across both eras is *statistical heterogeneity* (non-IID data): clients hold data drawn from different distributions, making a single shared model suboptimal for any individual client [Li *et al.*, 2022; Kairouz *et al.*, 2021]. PFL resolves this tension by letting each client retain a customized model while still benefiting from cross-client collaboration. The classical FL literature [McMahan *et al.*, 2017; Li *et al.*, 2020] already shows that the naive FedAvg strategy — average all parameters from all clients with equal weight — fails under heterogeneity, motivating a richer design space.

In this survey, we organize the entire PFL landscape around three fundamental and *orthogonal* dimensions of knowledge sharing:

Q1: What to share? Which knowledge representations are exchanged — full model parameters [McMahan *et al.*, 2017], partial parameters [Arivazhagan *et al.*, 2019; Collins *et al.*, 2021], class prototypes [Tan *et al.*, 2022], intermediate representations [Zhang *et al.*, 2023c], distillation outputs [Lin *et al.*, 2020], or LoRA adapters [Zhang *et al.*, 2023e]? In the LLM era, this question translates to: which components of a LoRA adapter (A, B, or both) should be shared, and whether a lightweight proxy model can substitute the full LLM on resource-constrained clients.

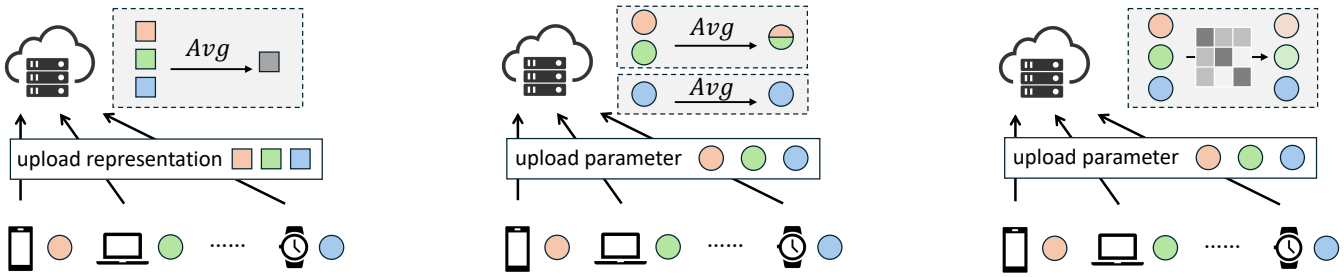
Q2: Who to share with? Which clients should collaborate? Naive aggregation over all participants introduces gradient conflicts [Sattler *et al.*, 2020]; clustering methods [Ghosh *et al.*, 2020; Liu *et al.*, 2024] or soft probabilistic assignments [Marfoq *et al.*, 2021] confine collaboration to compatible partners. In the LLM era, task-level similarity replaces data-distribution similarity as the primary criterion for partner selection [Zhou *et al.*, 2024].

Q3: How to share? By what mechanism is shared knowledge combined? The answer ranges from similarity-weighted aggregation [Huang *et al.*, 2021; Ye *et al.*, 2023] to layer-wise adaptive fusion [Ma *et al.*, 2022] and graph-propagated knowledge [Chen *et al.*, 2022b]. In the LLM era, the structural properties of LoRA ($\Delta W = BA$) demand new aggregation protocols that avoid mathematical noise introduced by naive averaging [Wang *et al.*, 2024].

These three dimensions apply uniformly across the model-scale FL era *and* the LLM-scale FL era, providing a stable conceptual scaffold for our revisit.

Contributions. *(i)* We propose a unified three-dimensional taxonomy (What / Who / How to share) that organizes 70 representative PFL methods into a coherent framework; *(ii)* We systematically cover the classical PFL literature (partial

*This is a technique report of my understanding to personalized federated learning. Version 2026/03/13.



(a) What to share? (b) Who to share with? (c) How to share?

Figure 1: The three-dimensional knowledge-sharing taxonomy unifying classical PFL and federated LLM fine-tuning.

parameters, prototypes, representations, distillation, clustering, graph aggregation) and identify the design principles that have proven most durable; *(iii)* We extend the same taxonomy to federated LLM fine-tuning via LoRA adapters, showing how each classical dimension acquires new, LLM-specific instantiations; *(iv)* We discuss benchmarks, open challenges, and future directions with particular attention to the tensions that become acute in the LLM era.

Contributions. In this survey, we: *(i)* propose a unified three-dimensional taxonomy (What / Who / How to share) that organizes the diverse PFL literature into a coherent framework; *(ii)* systematically review representative methods under this framework, covering traditional model-based FL and recent extensions; *(iii)* extend the taxonomy to the emerging area of federated large language model (LLM) fine-tuning via LoRA-based adapters; *(iv)* discuss benchmark datasets, evaluation protocols, and open research challenges with pointers to future directions.

2 Preliminaries

Federated Learning. In federated learning, N clients with local datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ collaboratively train a *single shared model* θ without sharing raw data, optimizing:

$$\min_{\theta} \sum_{i=1}^{N} p_i \, \mathcal{L}(\theta; \mathcal{D}_i), \tag{1}$$

where $p_i = |\mathcal{D}_i| / \sum_j |\mathcal{D}_j|$ is the relative data weight and $\mathcal{L}(\cdot; \mathcal{D}_i)$ is the local empirical loss of client i. FedAvg [McMahan *et al.*, 2017] optimizes Eq. (1) by alternating between multiple rounds of local SGD and server-side weighted averaging: $\theta \leftarrow \sum_i p_i \theta_i$. Under *statistical heterogeneity* (non-IID), where local data distributions $\mathcal{P}_i(\mathbf{x}, y)$ differ across clients, this simple averaging suffers from gradient divergence, impaired convergence, and poor per-client performance [Kairouz *et al.*, 2021; Li *et al.*, 2022].

Collaboration in Federated Learning. Let the collaboration matrix $W \in \mathbb{R}^{N \times N}$ specify how local models are combined for each client. The general PFL objective is:

$$\min_{\{W, \theta_i\}} \sum_{i=1}^{N} p_i \, \mathcal{L}\left(\sum_{j=1}^{N} W_{ij} \theta_j; \, \mathcal{D}_i \right) + \lambda \, \mathcal{R}(W; \boldsymbol{\theta}) \quad \text{s.t.} \quad \sum_{j=1}^{N} W_{ij} = 1, \; \forall i; \quad W_{ij} \geq 0, \; \forall i, j, \tag{2}$$

where $\mathcal{R}(W; \boldsymbol{\theta})$ regularizes the collaboration weights. The standard FedAvg objective Eq. (1) is recovered with $W_{ij} = p_j, \, \forall j$ and $\lambda = 0$.

Problem Decomposition. The design of an effective collaboration mechanism under Eq. (2) decomposes into three orthogonal aspects corresponding to our three survey dimensions.

- *What to share (Collaboration scope).* Let $\theta_i = \{\theta_i^{(l)}\}_{l=1}^{L}$ denote parameters across L components (layers or modules). The scope defines which subset $\mathcal{S}_i \subseteq \{1, \ldots, L\}$ participates in aggregation:

$$\theta_i^{(l)} \leftarrow \sum_{j=1}^{N} W_{ij}^{(l)} \, \theta_j^{(l)}, \quad l \in \mathcal{S}_i, \tag{3}$$

  while parameters outside $\mathcal{S}_i$ are updated purely from local data. Methods in this dimension vary from sharing full parameters [McMahan *et al.*, 2017] to sharing only partial layers [Arivazhagan *et al.*, 2019], prototypes [Tan *et al.*, 2022], or distillation outputs [Lin *et al.*, 2020].

- *Who to share with (Collaborator selection).* The active collaborator set for client i is $\mathcal{N}_i = \{j \mid W_{ij} > 0\}$, defining the nonzero sparsity pattern of W. $\mathcal{N}_i$ can be all clients [McMahan *et al.*, 2017], a fixed cluster [Ghosh *et al.*, 2020], or adaptively inferred from similarity $s_{ij} = \mathrm{sim}(\theta_i, \theta_j)$ [Sattler *et al.*, 2020].

- *How to share (Weight allocation).* Given $\mathcal{N}_i$, weight allocation specifies the magnitude of W_{ij}:

$$W_{ij} = \frac{\phi_{ij}}{\sum_{k \in \mathcal{N}_i} \phi_{ik}}, \quad j \in \mathcal{N}_i, \tag{4}$$

  where the collaboration score ϕ_{ij} may depend on data size [McMahan *et al.*, 2017], model similarity [Huang *et al.*, 2021], graph structure [Ye *et al.*, 2023], or a learned function [Shamsian *et al.*, 2021].

Model Heterogeneity. Beyond data heterogeneity, clients may use structurally different neural network architectures due to varying computational budgets. Prototype- and distillation-based methods [Tan *et al.*, 2022; Lin *et al.*, 2020] handle this naturally by sharing architecture-agnostic fixed-size vectors, while parameter-sharing methods [Arivazhagan *et al.*, 2019; Collins *et al.*, 2021] require compatible architectures.

Taxonomy Overview. Table. 1 summarizes representative methods along the three dimensions. The following sections review each dimension in detail, covering 68 non-LLM methods and 10 federated LLM fine-tuning methods.

3 What to Share: Knowledge Representation

The choice of *what* knowledge to exchange is the most fundamental design decision in PFL, as it determines both the expressiveness of shared information and what heterogeneity can be tolerated. We identify four sub-categories: partial model parameters, class prototypes, intermediate representations, and knowledge distillation outputs.

3.1 Partial Parameter Sharing

The most direct approach to personalization splits the model into a *shared part* (aggregated globally) and a *private part* (kept locally), exploiting the observation that different layers encode different levels of abstraction.

Canonical Layer Splits. FedPer [Arivazhagan *et al.*, 2019] partitions the network into shared base layers and local personalization layers (typically the final fully-connected heads), where the base captures cross-client feature commonalities while the head adapts to local class distributions. In contrast, **LG-FedAvg** [Liang *et al.*, 2020] reverses this intuition: higher (more abstract) layers are shared globally while lower layers remain local, on the premise that high-level semantic representations are more transferable. **FedBN** [Li *et al.*, 2021b] takes a minimal-intervention approach, keeping only Batch Normalization (BN) statistics local while aggregating all other parameters. Since BN parameters encode dataset-specific feature statistics, this small change effectively handles feature shift caused by non-IID data.

Representation–Head Decoupling. FedRep [Collins *et al.*, 2021] provides theoretical grounding for sharing only the feature extractor (body) and learning a local classifier head. The local head is optimized with multiple local gradient steps each round while the body follows a single global update. **FedBABU** [Oh *et al.*, 2022] freezes the head during federated rounds and performs post-FL fine-tuning, while **FedRoD** [Chen and Chao, 2022] jointly maintains a global head (for generalization) and a local head (for personalization), blending both at inference. **Ditto** [Li *et al.*, 2021a] takes a complementary angle: each client maintains both a global FedAvg model and a personalized local model, regularized toward the global model by a proximal term, providing provable per-client fairness and robustness guarantees. **FedAIMS** [Li *et al.*, 2025] further introduces adaptive intermediate supervision signals to align the learned representations at the split point.

Data-Driven Parameter Selection. FedSelect [Tamirisa *et al.*, 2024] abandons pre-defined layer boundaries by borrowing the Lottery Ticket Hypothesis: it progressively identifies each client's personalized subnetwork, retaining locally the parameters with large gradient magnitudes (task-critical) and sharing small-magnitude parameters globally. This fully data-driven split requires no prior architectural knowledge.

3.2 Prototype-based Sharing

Prototype-based methods share *class-level feature averages* (prototypes) rather than model parameters. Prototypes are architecture-agnostic fixed-size vectors, making this family naturally suited to model-heterogeneous settings.

FedProto [Tan *et al.*, 2022] establishes the paradigm: each client computes per-class feature prototypes and uploads them; the server aggregates global prototypes used to regularize local training. Building on it, **FedProc** [Wu *et al.*, 2023] introduces supervised contrastive learning, using global prototypes as anchors to pull same-class embeddings together and push cross-class embeddings apart, alleviating representation drift under severe non-IID conditions. **FedPHP** [Yi *et al.*, 2023] augments prototype hints with a private model inheritance mechanism to prevent catastrophic forgetting when adapting to global prototype signals.

FedTGP [Zhang *et al.*, 2024a] observes that simple prototype averaging can undermine inter-class discriminability. It maintains *trainable* global prototypes on the server, optimized via adaptive-margin contrastive loss to improve quality without relying on averaging. **FedSSA** [Yi *et al.*, 2024] shifts from feature prototypes to classifier header sharing, measuring similarity by semantic embeddings of classifier weight vectors and selectively aggregating headers from similar clients. **FedCPD** [Wu and others, 2025] further integrates prototype-driven dynamic aggregation weights, combining prototype similarity with a contrastive loss for sharper personalized representations.

3.3 Representation-based Sharing

These methods share intermediate feature representations to align cross-client feature spaces while retaining personalized decision boundaries. **FedCR** [Zhang *et al.*, 2023b] employs conditional mutual information (CMI) regularization to drive cross-client representation alignment without any explicit parameter exchange. **GPFL** [Zhang *et al.*, 2023c] explicitly decouples the extractor into a shared global branch and a private local branch trained jointly, with a feature similarity constraint pushing the global branch toward cross-client generic attributes.

3.4 Knowledge Distillation-based Sharing

Knowledge distillation (KD) methods share model *outputs* — soft labels, logits, or dark knowledge — rather than parameters. This is inherently model-heterogeneous, as only prediction vectors are exchanged.

FedDF [Lin *et al.*, 2020] performs server-side ensemble distillation on a public unlabeled dataset: client models produce soft predictions that serve as a teacher ensemble for distilling a global student model. **FedBE** [Chen and Chao, 2021]

Table 1: Representative personalized FL methods organized by the three-dimensional knowledge-sharing taxonomy. *What*: Full = full model parameters; Partial = partial/decoupled parameters; Proto = class prototypes; KD = knowledge distillation outputs. *Who*: All = all clients; Hard = hard cluster; Soft = soft cluster membership. *How*: Avg = weighted averaging; Prox = proximal regularization; Weighted = similarity-weighted; HyperNet = hypernetwork-generated; Graph = graph-propagated.

Method	**Venue**	**Q1: What**	**Q2: Who**	**Q3: How**	**Model-Hetero?**
Baseline					
FedAvg [McMahan *et al.*, 2017]	AISTATS'17	Full	All	Avg	×
FedProx [Li *et al.*, 2020]	MLSys'20	Full	All	Prox	×
What to Share — Partial Parameter Splitting					
FedPer [Arivazhagan *et al.*, 2019]	arXiv'19	*Partial (base layers)*	All	Avg	×
LG-FedAvg [Liang *et al.*, 2020]	arXiv'20	*Partial (top layers)*	All	Avg	×
FedBN [Li *et al.*, 2021b]	ICLR'21	*Partial (non-BN)*	All	Avg	×
FedRep [Collins *et al.*, 2021]	ICML'21	*Partial (encoder)*	All	Avg	×
Ditto [Li *et al.*, 2021a]	ICML'21	Full	All	Prox	×
FedSelect [Tamirisa *et al.*, 2024]	CVPR'24	*Partial (data-driven)*	All	Avg	×
What to Share — Prototype-based					
FedProto [Tan *et al.*, 2022]	AAAI'22	*Proto*	All	Avg	✓
FedProc [Wu *et al.*, 2023]	FGCS'23	*Proto + Contrastive*	All	Avg	✓
FedPHP [Yi *et al.*, 2023]	ECML'23	*Proto (hints)*	All	Avg	✓
FedTGP [Zhang *et al.*, 2024a]	AAAI'24	*Proto (trainable)*	All	Avg	✓
FedSSA [Yi *et al.*, 2024]	IJCAI'24	*Proto (classifier)*	All	*Weighted*	✓
FedCPD [Wu and others, 2025]	IJCAI'25	*Proto + Dynamic*	All	*Weighted*	✓
What to Share — Representation-based					
FedCR [Zhang *et al.*, 2023b]	ICML'23	*Representation*	All	Avg	×
GPFL [Zhang *et al.*, 2023c]	ICCV'23	*Dual Feature*	All	Avg	×
What to Share — Knowledge Distillation					
FedDF [Lin *et al.*, 2020]	NeurIPS'20	*KD (logits)*	All	*Distillation*	✓
FedBE [Chen and Chao, 2021]	ICLR'21	*KD (Bayesian)*	All	*Distillation*	✓
FedGen [Zhu *et al.*, 2021]	ICML'21	*KD (generated)*	All	*Distillation*	✓
DENSE [Zhang *et al.*, 2022]	NeurIPS'22	*KD (one-shot)*	All	*Distillation*	✓
CAFEDistill [Liu *et al.*, 2026]	arXiv'26	*KD (early-exit)*	All	*Distillation*	✓
Who to Share With — Hard Clustering					
CFL [Sattler *et al.*, 2020]	TNNLS'21	Full	*Hard*	Avg	×
IFCA [Ghosh *et al.*, 2020]	NeurIPS'20	Full	*Hard*	Avg	×
FL+HC [Briggs *et al.*, 2020]	IJCNN'20	Full	*Hard*	Avg	×
FedGroup [Duan *et al.*, 2021]	ISPA'21	Full	*Hard*	Avg	×
FLIS [Morafah *et al.*, 2023]	TAI'23	Full	*Hard (logit)*	Avg	✓
PACFL [Vahidian *et al.*, 2023]	AAAI'23	Full	*Hard (SVD)*	Avg	×
Dennis *et al.* [Dennis *et al.*, 2021]	ICML'21	Full	*Hard (one-shot)*	Avg	×
CASA [Liu *et al.*, 2024]	KDD'24	Full	*Hard (async)*	*Weighted (staleness)*	×
Who to Share With — Soft Clustering					
FedEM [Marfoq *et al.*, 2021]	NeurIPS'21	Full	*Soft (EM)*	*Weighted*	×
FedSoft [Ruan and Joe-Wong, 2022]	AAAI'22	Full	*Soft*	Prox	×
FedRC [Liu *et al.*, 2023]	ICML'23	Full	*Soft (robust)*	*Weighted*	×
FeSEM [Long *et al.*, 2023]	WWW'23	Full	*Soft (SEM)*	*Weighted*	×
FedMoM [Ghari and Shen, 2024]	NeurIPS'24	Full	*Soft (online)*	*Weighted*	×
How to Share — Pairwise Similarity-Weighted					
FedFomo [Zhang *et al.*, 2021]	ICLR'21	Full	All	*Weighted (1st-order)*	×
FedAMP [Huang *et al.*, 2021]	AAAI'21	Full	All	*Weighted (attention)*	×
APPLE [Luo *et al.*, 2022]	IJCAI'22	Full	All	*Weighted (learned)*	×
FedSim [Palihawadana *et al.*, 2022]	Neurocomp.'22	Full	All	*Weighted (gradient)*	×
FedDWA [Li *et al.*, 2023]	IJCAI'23	Full	All	*Weighted (dynamic)*	×
pFedSim [Tan *et al.*, 2023]	arXiv'23	*Partial (encoder)*	All	*Weighted (classifier)*	×
pFedGraph [Ye *et al.*, 2023]	ICML'23	Full	All	*Weighted (QP graph)*	×
AsyncPFL [Anonymous, 2025]	arXiv'25	Full	All	*Weighted (staleness-aware)*	×
How to Share — Local–Global Adaptive					
pFedLA [Ma *et al.*, 2022]	CVPR'22	Full	All	*HyperNet (per-layer)*	×
FedALA [Zhang *et al.*, 2023a]	AAAI'23	Full	All	*Weighted (element-wise)*	×
FedPAC [Xu *et al.*, 2023]	ICLR'23	*Partial (classifier)*	All	*Weighted (theory)*	×
KNN-Per [Marfoq *et al.*, 2022]	ICML'22	Full	All	*Weighted (kNN)*	×
How to Share — Learnable Graph-driven					
SFL [Chen *et al.*, 2022b]	IJCAI'22	Full	All	*Graph (GCN)*	×
Fed-GNN [Xia *et al.*, 2022]	NatComm.'22	Full	All	*Graph (GNN)*	×
FedAGHN [Xu and others, 2025]	arXiv'25	Full	All	*Graph + HyperNet*	×
pFedGAT [Ye and others, 2025]	arXiv'25	Full	All	*Graph (GAT)*	×

replaces the simple ensemble with Bayesian model ensemble — the server samples high-quality global models from a Gaussian mixture fit to client model distributions, achieving better out-of-distribution robustness. To remove the dependency on a public dataset, **FedGen** [Zhu *et al.*, 2021] trains a lightweight server-side generator learning the latent distribution from client logit responses, while **DENSE** [Zhang *et al.*, 2022] extends this to one-shot FL via GAN-based pseudo-data generation followed by ensemble distillation in a single communication round.

CAFEDistill [Liu *et al.*, 2026] introduces Early-Exit Networks (EENs) into the FL distillation paradigm, enabling each client to dynamically choose its inference depth at test time — achieving both personalization and compute adaptability. The core contribution is a *progressive depth-prioritized student coordination* mechanism that distills knowledge across multiple client exits while counteracting the depth-wise interference between shallow and deep exits caused by heterogeneous data distributions. Compared to existing distillation baselines, CAFEDistill reduces inference cost by 30–47% without accuracy loss.

4 Who to Share With: Client Selection and Clustering

Sharing knowledge indiscriminately with all clients introduces conflicting gradients from incompatible data distributions. The *Who to Share With* dimension determines which clients collaborate, confining knowledge exchange to compatible partners.

4.1 Hard Clustering

Hard clustering partitions clients into disjoint groups; aggregation only occurs within each group.

CFL [Sattler *et al.*, 2020] uses cosine similarity between client gradient vectors to detect distribution conflicts, recursively applying a bipartite split to form a hierarchy of clusters. **IFCA** [Ghosh *et al.*, 2020] maintains K global models on the server; each round, clients evaluate all K models on local data, join the cluster with minimum loss, and update that cluster's model, with convergence guarantees. **FL+HC** [Briggs *et al.*, 2020] performs hierarchical agglomerative clustering on local update vectors before global aggregation. **FedGroup** [Duan *et al.*, 2021] designs a decomposed data-driven similarity measure combining gradient direction and data statistics for communication-efficient clustering. **FLIS** [Morafah *et al.*, 2023] determines cluster assignment by comparing inference logits on a shared public dataset, avoiding direct parameter exposure. **PACFL** [Vahidian *et al.*, 2023] uses principal angles between client data subspaces (computed via truncated SVD) as a geometric distribution similarity metric. **Dennis *et al.*** [Dennis *et al.*, 2021] show that one-shot clustering is achievable by exploiting the statistical heterogeneity of client models themselves.

CASA [Liu *et al.*, 2024] extends clustered FL to the practically important *asynchronous* setting, where clients upload model updates at different times and a global synchronization barrier is prohibitive. It proposes a buffer-aided dynamic clustering algorithm that computes pairwise cosine similarity as updates arrive in the buffer, and introduces staleness-aware training weights to mitigate the adverse effects of stale gradients. A bi-level asynchronous aggregation strategy coordinates intra-cluster (synchronous) and inter-cluster (asynchronous) updates, achieving 2.3–6.5$\times$ speedup over synchronous clustered FL baselines at comparable accuracy.

4.2 Soft Clustering

Soft clustering allows fractional memberships across multiple clusters, accommodating mixed or uncertain data distributions.

FedEM [Marfoq *et al.*, 2021] models each client's data distribution as a mixture of K component distributions and uses the EM algorithm to alternately optimize soft membership weights (E-step) and component models (M-step). **FedSoft** [Ruan and Joe-Wong, 2022] relaxes hard membership to continuous weights, using proximal local updates to balance consistency across component models with local personalization, with theoretical convergence guarantees. **FedRC** [Liu *et al.*, 2023] addresses the realistic scenario where clients simultaneously experience both feature and label distribution shifts, proposing a robust soft clustering framework with theoretical bounds. **FeSEM** [Long *et al.*, 2023] frames multi-center FL using structural equation model iterations for joint cluster-assignment and model optimization. **FedMoM** [Ghari and Shen, 2024] adopts an online learning perspective where each client's prediction is a dynamically-weighted mixture of a fine-tuned local model and server-maintained global models, adapting to non-stationary distribution changes without pre-specified cluster counts.

5 How to Share: Aggregation Mechanisms

Given shared knowledge (What) and a collaboration scope (Who), the aggregation mechanism (How) determines the weights and procedures by which knowledge is combined into each client's personalized model. We identify three subcategories: pairwise similarity-weighted collaboration, local–global adaptive aggregation, and learnable graph-driven aggregation.

5.1 Pairwise Similarity-Weighted Collaboration

These methods compute a personalized aggregation weight W_{ij} for each peer model received by client i, based on an estimated similarity between client i and client j.

FedFomo [Zhang *et al.*, 2021] estimates each received model's marginal utility on the local validation set via a first-order Taylor approximation, automatically assigning large weights to similar clients and down-weighting conflicting ones. **FedAMP** [Huang *et al.*, 2021] uses an attention-inducing function on the server to compute pairwise collaboration strength from model parameter similarities, enabling more information to flow between model-similar clients. **APPLE** [Luo *et al.*, 2022] learns per-client trainable weight vectors over all other clients' models through MAML-style bi-level optimization. **FedSim** [Palihawadana *et al.*, 2022] hierarchically applies gradient-similarity-based local aggregation within subgroups, then global aggregation, reducing gradient variance. **FedDWA** [Li *et al.*, 2023] formulates weight

computation as a server-side optimization minimizing the distance between each personalized model and a guidance model derived from local gradient directions. **pFedSim** [Tan *et al.*, 2023] decouples the model into extractor and classifier, uses classifier similarity as the collaboration metric, and aggregates only extractors — protecting the local classifier while sharing cross-client features.

pFedGraph [Ye *et al.*, 2023] elevates pairwise similarity weighting to a theoretically grounded framework: the server infers optimal collaboration graph edge weights by solving a quadratic programming (QP) problem with pairwise model similarity and dataset size constraints. This formulation subsumes many hand-crafted similarity aggregation schemes and provides a convergence guarantee that the resulting personalized models outperform FedAvg.

AsyncPFL [Anonymous, 2025] extends pairwise similarity-weighted collaboration to the *asynchronous* setting, where the staleness gap between uploaded models causes cosine similarity to become unreliable. It proposes a staleness-aware similarity correction mechanism that preserves the semantic validity of pairwise weights while accommodating asynchronous communication protocols, making similarity-based PFL practical in mobile and edge deployment environments.

5.2 Local–Global Adaptive Aggregation

Rather than aggregating horizontally across clients, local–global methods vertically combine a global federated model and a local personalized model for each client, learning how much of each to incorporate at different levels of granularity.

pFedLA [Ma *et al.*, 2022] trains a hypernetwork on the server that generates *per-layer* aggregation weights for each client independently. Shallow layers (encoding more local, dataset-specific features) receive less global influence than deep layers (encoding more transferable semantic representations), achieving fine-grained personalized aggregation without manual specification. **FedALA** [Zhang *et al.*, 2023a] performs client-side adaptive aggregation: before each local training round, the client learns element-wise weights for selectively incorporating useful parameters from the received global model, replacing the simple global model overwrite through a local optimization step.

FedPAC [Xu *et al.*, 2023] theoretically derives the optimal classifier aggregation weights as a function of inter-client feature distribution differences, providing a principled local–global mechanism for classifier collaboration while aligning feature extractors via contrastive regularization. **KNN-Per** [Marfoq *et al.*, 2022] replaces client-level aggregation weights with *sample-level* similarity, augmenting global model predictions with k-nearest-neighbor retrieval in the shared representation space — achieving fine-grained local memorization beyond what cluster- or client-level aggregation can provide.

5.3 Learnable Graph-driven Aggregation

Graph-driven methods model inter-client relationships as a weighted directed graph with learnable edge weights, propagating knowledge along graph edges via graph neural network (GNN) message-passing mechanisms. Unlike pairwise methods that treat edges as fixed scalars, graph methods capture higher-order neighborhood relationships and co-optimize the graph structure with the client models.

SFL [Chen *et al.*, 2022b] constructs a client relationship graph from model parameter similarities with trainable edge weights, and uses GCN-style message passing to aggregate neighbor models into personalized initializations for the next round — jointly learning graph structure and model aggregation. **Fed-GNN** [Xia *et al.*, 2022] builds a privacy-preserving inter-client graph and propagates user feature representations using a federated GNN, particularly suited for recommendation scenarios where client relationships carry semantic meaning.

FedAGHN [Xu and others, 2025] combines attentive graph neural networks with hypernetworks to generate per-layer, per-client aggregation weights: the graph attention mechanism captures global cooperation topology while the hypernetwork customizes layer-wise weights for each client, achieving fine-grained personalized aggregation at both graph and layer granularities. **pFedGAT** [Ye and others, 2025] uses graph attention networks to dynamically learn the collaboration graph structure during training, with attention weights adapting to non-stationary distribution shifts across rounds — enabling the topology to evolve as client data distributions change over time.

6 Federated LLM Fine-tuning

The rise of large language models (LLMs) has opened a new frontier in PFL: federated fine-tuning of LLMs under heterogeneous task distributions while preserving data privacy. Parameter-efficient fine-tuning (PEFT) methods, especially Low-Rank Adaptation (LoRA) [Kairouz *et al.*, 2021], make federated LLM fine-tuning practical by drastically reducing the number of trainable parameters. In LoRA, each weight update $\Delta W \in \mathbb{R}^{d\times k}$ is approximated by two low-rank matrices as $\Delta W = BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and rank $r \ll \min(d, k)$.

The What / Who / How taxonomy naturally extends to federated LoRA fine-tuning, with the shared objects being LoRA *adapters* rather than full model parameters. This section reviews 10 representative methods across the three dimensions.

6.1 What to Share in LoRA Fine-tuning

FedIT [Zhang *et al.*, 2023e] establishes the baseline for federated instruction tuning by applying FedAvg directly to LoRA matrices: clients train LoRA adapters locally and the server independently averages the A and B matrices. However, this introduces aggregation noise because different clients initialize A matrices differently, meaning $\overline{B}\cdot\overline{A} \neq \overline{BA}$ in general.

FFA-LoRA [Han *et al.*, 2024] resolves this by fixing all clients' A matrices to the same random initialization and fine-tuning only B matrices. Since all A matrices are identical, averaging only B yields noise-free aggregation equivalent to directly combining the full low-rank updates $\Delta W = B\bar{A}$.

FedDPA [Yang *et al.*, 2024] and **FDLoRA** [Qi *et al.*, 2024] adopt dual-adapter designs: a global LoRA adapter is shared

and aggregated on the server (capturing cross-client generalizable knowledge), while a local LoRA adapter is kept private (preserving client-specific knowledge). At inference, both adapters are composed additively. FDLoRA additionally introduces a fusion optimizer that jointly updates both adapters' gradients to prevent the "bucket effect", where overall performance is dominated by the weakest client, and supports heterogeneous LoRA ranks across clients.

FedBiOT [Zhang *et al.*, 2024b] addresses memory constraints: clients may not hold the full LLM. A compact proxy model is distilled from the full LLM on the server and sent to clients; clients fine-tune lightweight LoRA adapters on the proxy model and upload them. The server transfers adapter knowledge back to the full LLM via bi-level optimization, dramatically reducing client-side memory and communication requirements.

6.2 Who to Share With in LoRA Fine-tuning

FedAMoLE [Zhou *et al.*, 2024] applies a Mixture-of-Experts (MoE) framework where each client dynamically selects from multiple domain-specific LoRA experts based on task similarity. Clients with similar tasks share the same LoRA expert, enabling selective collaboration while supporting heterogeneous LoRA ranks across clients — expert assignment is driven entirely by data-driven similarity estimation.

FedLEASE [Fu and others, 2025] begins with a preliminary training phase that determines the optimal cluster count via the silhouette coefficient, then clusters clients by LoRA parameter similarity using this data-adaptive criterion. Each cluster trains a dedicated LoRA expert, and at inference, each client routes to its most similar expert, enabling fine-grained task-aware personalized prediction.

6.3 How to Share LoRA Adapters

FLoRA [Wang *et al.*, 2024] proposes *stacking aggregation* to replace independent A/B averaging. Client LoRA matrices are concatenated along the rank dimension with scaling coefficients, rather than averaged element-wise. This achieves noise-free aggregation and naturally supports heterogeneous ranks: the aggregated global LoRA has rank equal to the sum of all client ranks, directly avoiding the $\overline{B} \cdot \overline{A}$ inconsistency of FedIT.

FlexLoRA [Bai *et al.*, 2024] supports dynamic LoRA ranks adaptive to heterogeneous client computational resources. During aggregation, each client projects its local LoRA to singular value decomposition, and re-aggregation proceeds rank-by-rank. The server maintains a global LoRA of higher rank than any individual client's, alleviating the bucket effect from rank heterogeneity and enabling clients with limited resources to still contribute meaningfully.

FedALT [Koita and others, 2025] maintains both an individual LoRA (for local personalization) and a Rest-of-World (RoW) LoRA (aggregating knowledge from all other clients) per client. An input-conditioned adaptive mixer, inspired by MoE token-level routing, dynamically determines the mixing ratio of the two adapters at inference time. This fine-grained conditional blending achieves superior balance between global knowledge and local personalization compared to static adapter combination strategies.

7 Benchmarks and Evaluation

PFLlib [Zhang *et al.*, 2023d]. PFLlib is a beginner-friendly and comprehensive PFL library and benchmark designed to lower the barrier to entry for PFL research. It provides clean, modular implementations of 30+ PFL algorithms with a unified interface, enabling researchers to reproduce baselines and prototype new methods with minimal boilerplate. PFLlib covers diverse non-IID partitioning strategies — including Dirichlet label skew, pathological partitioning, and feature shift — and supports both cross-device and cross-silo settings. Its focus on code readability and extensibility makes it particularly accessible to newcomers, while the integrated benchmark suite facilitates fair and reproducible comparisons across methods.

pFL-Bench [Chen *et al.*, 2022a]. pFL-Bench provides the most comprehensive PFL-specific benchmark to date, covering 10+ datasets across vision and NLP tasks with standardized implementations of 20+ PFL methods — including the majority of methods reviewed in this survey. It systematically evaluates methods along multiple dimensions: generalization, fairness (variance in per-client performance), communication efficiency, and convergence speed. The benchmark is built on the FederatedScope framework, enabling easy integration of new methods.

Evaluation Dimensions. Beyond raw per-client accuracy, a thorough PFL evaluation should consider: (1) *Generalization*: performance on clients unseen during training, especially important for hypernetwork-based methods; (2) *Fairness*: the variance and worst-case performance across clients, not just average accuracy; (3) *Communication cost*: total bytes transmitted, critical for practical deployment; (4) *Computational overhead*: local computation per round, particularly relevant for resource-heterogeneous settings; (5) *Model heterogeneity support*: whether the method accommodates different architectures across clients, as tested by prototype- and distillation-based methods.

8 Discussion and Future Directions

Privacy–Utility Trade-off. The choice of what to share directly impacts privacy. Sharing full model parameters enables parameter reconstruction attacks, while sharing only prototypes [Tan *et al.*, 2022] or distillation outputs [Lin *et al.*, 2020] reduces exposure but sacrifices expressiveness. Designing sharing mechanisms with formal differential privacy guarantees while minimizing the utility loss remains an open challenge. Secure aggregation protocols [Kairouz *et al.*, 2021] can complement similarity-based methods but may conflict with the need to compute pairwise client similarities.

Communication Efficiency. Several high-performing methods incur communication overhead exceeding FedAvg. pFedLA [Ma *et al.*, 2022] requires the server to maintain per-client hypernetworks; FedFomo [Zhang *et al.*, 2021] requires clients to download all peer models; graph-based methods [Ye *et al.*, 2023] must construct and communicate graph structures. Quantization, sparsification, and one-shot strategies [Dennis *et al.*, 2021; Zhang *et al.*, 2022] are complementary solutions, but their joint integration with similarity-based collaboration deserves further investigation.

Scalability of Graph and Similarity Methods. Graph-based methods model pairwise client relationships, incurring $O(N^2)$ complexity that becomes prohibitive at scale. Sparse graph approximations, random walk-based propagation [Ye *et al.*, 2023], and hierarchical decomposition [Sattler *et al.*, 2020] partially address this, but scalability to thousands of clients in cross-device FL remains largely unsolved.

Non-stationary Environments. Most PFL methods assume static data distributions. Yet real-world deployments commonly experience concept drift and distribution shifts over time. FedMoM [Ghari and Shen, 2024] and pFedGAT [Ye and others, 2025] partially address non-stationarity — the former via online mixture adaptation and the latter through time-evolving attention graph topology — but comprehensive solutions that detect, adapt to, and recover from arbitrary distribution dynamics remain underexplored.

Similarity vs. Complementarity. Most similarity-based methods exclusively collaborate with similar clients. However, recent work [Wu *et al.*, 2025] suggests that *dissimilar* clients can also provide valuable complementarity: diverse knowledge transfer can improve generalization beyond what homogeneous collaboration achieves. How to optimally balance similarity-based homogeneity and diversity-based complementarity in a principled way is an emerging research direction.

Theoretical Foundations. While IFCA [Ghosh *et al.*, 2020], FedSoft [Ruan and Joe-Wong, 2022], and pFedGraph [Ye *et al.*, 2023] provide convergence analyses, many state-of-the-art PFL methods lack formal guarantees. Establishing the conditions under which similarity-based collaboration strictly improves over independent local training, and quantifying the benefit as a function of inter-client distribution similarity, are important open theoretical questions.

Federated LLM-specific Challenges. Federated LLM fine-tuning inherits the above challenges but introduces unique additional difficulties: (1) *Rank heterogeneity*: clients with different resources use different LoRA ranks, requiring new aggregation protocols (FLoRA [Wang *et al.*, 2024], FlexLoRA [Bai *et al.*, 2024]); (2) *Catastrophic forgetting*: incorporating global adapters can overwrite local knowledge, motivating dual-adapter designs [Yang *et al.*, 2024; Qi *et al.*, 2024]; (3) *Instruction diversity*: different clients may hold instructions from vastly different domains, exacerbating the standard non-IID problem; (4) *Evaluation*: standard NLP benchmarks are designed for centralized settings; federated evaluation protocols for instruction-tuned LLMs are still nascent.

9 Conclusion

We have presented a systematic survey of personalized federated learning from the perspective of knowledge sharing. Our three-dimensional taxonomy — *What to Share*, *Who to Share With*, and *How to Share* — provides a unified framework that organizes severak representative methods spanning prototype-based learning, client clustering, graph aggregation, hypernetworks, and federated LLM fine-tuning.

Several key insights emerge from this review. First, the choice of knowledge representation (What) fundamentally determines what heterogeneity can be accommodated: prototype and distillation methods naturally handle model-architectural heterogeneity, while parameter-sharing methods require compatible architectures. Second, hard clustering offers strong personalization but depends critically on accurate similarity estimation, while soft clustering provides graceful degradation when data distributions are ambiguous or mixed. Third, graph-based and hypernetwork aggregation methods offer the richest modeling of inter-client relationships but at increased computational and communication cost. Fourth, the What / Who / How taxonomy naturally extends to federated LLM fine-tuning, where LoRA adapters replace full parameters as the shared object, introducing new challenges around rank heterogeneity and adapter composition.

Looking ahead, we anticipate advances in privacy-preserving similarity estimation, scalable graph construction methods, and theoretically grounded aggregation mechanisms to drive the field forward. The federated LLM fine-tuning frontier, where massive model scale meets heterogeneous client tasks, presents particularly rich opportunities for extending the ideas reviewed in this survey.

References

[Anonymous, 2025] Anonymous. Asyncpfl: Staleness-aware similarity correction for asynchronous personalized federated learning. *arXiv preprint*, 2025.

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

[Bai *et al.*, 2024] Jiamu Bai, Daoyuan Chen, Bingchen Qian, Bolin Ding, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *NeurIPS*, 2024.

[Briggs *et al.*, 2020] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *IJCNN*, 2020.

[Chen and Chao, 2021] Hong-You Chen and Wei-Lun Chao. FedBE: Making Bayesian model ensemble applicable to federated learning. In *ICLR*, 2021.

[Chen and Chao, 2022] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*, 2022.

[Chen *et al.*, 2022a] Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. pFL-Bench: A comprehensive benchmark for personalized federated learning. In *NeurIPS*, 2022.

[Chen *et al.*, 2022b] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with a graph. In *IJCAI*, 2022.

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

[Dennis *et al.*, 2021] Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *ICML*, 2021.

[Duan *et al.*, 2021] Mingqian Duan, Dengyu Liu, Xinyuan Chen, Renping Liu, Yujing Tan, and Liang Liang. FedGroup: Efficient clustered federated learning via decomposed data-driven measure. In *IEEE ISPA*, 2021.

[Fu and others, 2025] Tianyu Fu et al. Adaptive LoRA experts allocation and selection for federated fine-tuning. *arXiv preprint*, 2025.

[Ghari and Shen, 2024] Pouya M Ghari and Yanning Shen. Personalized federated learning with mixture of models for adaptive prediction and model fine-tuning. In *NeurIPS*, 2024.

[Ghosh *et al.*, 2020] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *NeurIPS*, 33:19586–19597, 2020.

[Han *et al.*, 2024] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, et al. FFA-LoRA: Federated fine-tuning of language models with full freezing aggregation. In *ICLR*, 2024.

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

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

[Koita and others, 2025] Seydou Koita et al. FedALT: Federated fine-tuning through adaptive local training with rest-of-world LoRA. *arXiv preprint*, 2025.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

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

[Li *et al.*, 2021b] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *ICLR*, 2021.

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

[Li *et al.*, 2023] Jiahao Li, Enyuan Diao, Shaoming Weng, et al. FedDWA: Personalized federated learning with dynamic weight adjustment. In *IJCAI*, 2023.

[Li *et al.*, 2025] Shuyuan Li, Boyi Liu, Zimu Zhou, and Jin Dong. Fedaims: Adaptive intermediate supervision for personalized federated learning. *Frontiers of Computer Science*, 2025.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, 2020.

[Liu *et al.*, 2023] Yongxin Liu, Jiahao Shi, Jingjing Gu, Weichao Tao, Kaigui Bian, and Pengfei Wan. FedRC: Tackling diverse distribution shifts challenge in federated learning by robust clustering. In *ICML*, 2023.

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

[Liu *et al.*, 2026] Boyi Liu, Zimu Zhou, and Yongxin Tong. Cafedistill: Communication-efficient adaptive federated early-exit distillation. *arXiv preprint arXiv:2601.10015*, 2026.

[Long *et al.*, 2023] Guoqiang Long, Yue Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: Clients clustering for better personalization. *World Wide Web*, 2023.

[Luo *et al.*, 2022] Yan Luo, Yongkang Chen, Marios Savvides, Wei Chen, and Jinbo Bi. Adapt to adaptation: Learning personalization for cross-silo federated learning. In *IJCAI*, 2022.

[Ma *et al.*, 2022] Xiaoke Ma, Junpeng Zhang, Jinlong Zhang, Boshi Han, Ruichao Zhang, Mingkang Lin, and Fei Liu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, 2022.

[Marfoq *et al.*, 2021] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *NeurIPS*, 2021.

[Marfoq *et al.*, 2022] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *ICML*, 2022.

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

[Morafah *et al.*, 2023] Mahdi Morafah, Saeed Vahidian, Weijia Wang, and Bill Lin. FLIS: Clustered federated learning via inference similarity for non-IID data distribution. *IEEE Transactions on Artificial Intelligence*, 2023.

[Oh *et al.*, 2022] Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *ICLR*, 2022.

[Palihawadana *et al.*, 2022] Chamath Palihawadana, Nirmalie Wiratunga, Harsha Kalutarage, and Anjana Wijekoon. FedSim: Similarity guided model aggregation for federated learning. *Neurocomputing*, 507:157–168, 2022.

[Qi *et al.*, 2024] Jiaxing Qi, Zhongzhi Luo, Shaohan Huang, et al. FDLoRA: Personalized federated learning of large language model via dual LoRA tuning. *arXiv preprint arXiv:2406.07925*, 2024.

[Ruan and Joe-Wong, 2022] Yichen Ruan and Carlee Joe-Wong. FedSoft: Soft clustered federated learning with proximal local updating. In *AAAI*, 2022.

[Sattler *et al.*, 2020] 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.

[Shamsian *et al.*, 2021] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *ICML*, 2021.

[Tamirisa *et al.*, 2024] Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Li, Xingjun Yao, Elham Yao, Boyuan Yang, and Ce Chen. FedSelect: Personalized federated learning with customized selection of parameters for fine-tuning. In *CVPR*, 2024.

[Tan *et al.*, 2022] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. In *AAAI*, 2022.

[Tan *et al.*, 2023] Jiahao Tan, Guodong Long, Lu Liu, Tianyi Zhou, Jing Jiang, and Chengqi Zhang. pFedSim: Similarity-aware model aggregation towards personalized federated learning. *arXiv preprint arXiv:2305.15706*, 2023.

[Vahidian *et al.*, 2023] Saeed Vahidian, Mahdi Morafah, Weijia Chen, Uday Bhatt, Yang Liu, Mubarak Shah, and Bill Lin. Personalized federated learning via principal angles between subspaces. In *AAAI*, 2023.

[Wang *et al.*, 2024] Ziyao Wang, Zheyu Shi, Zhongyu Wang, Yanli Liu, et al. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *NeurIPS*, 2024.

[Wu and others, 2025] Xuting Wu et al. FedCPD: Personalized federated learning with prototype-enhanced dynamic aggregation. In *IJCAI*, 2025.

[Wu *et al.*, 2023] Xuting Wu, Dezhong Gao, and Guanghao Li. FedProc: Prototypical contrastive federated learning on non-IID data. *Future Generation Computer Systems*, 143:93–104, 2023.

[Wu *et al.*, 2025] Xinghao Wu, Jianwei Niu, Xuefeng Liu, Guogang Zhu, Shaojie Tang, Wanyu Lin, and Jiannong Cao. The diversity bonus: Learning from dissimilar clients in personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[Xia *et al.*, 2022] Rui Xia, Mang Ye, and Peng Hu. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):3194, 2022.

[Xu and others, 2025] Sheng Xu et al. FedAGHN: Personalized federated learning with attentive graph HyperNetworks. *arXiv preprint*, 2025.

[Xu *et al.*, 2023] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *ICLR*, 2023.

[Yang *et al.*, 2024] Yiyuan Yang, Guodong Luo, Xiaoyong Li, et al. Dual-personalizing adapter for federated foundation models. In *NeurIPS*, 2024.

[Ye and others, 2025] Rui Ye et al. Personalized federated learning via learning dynamic graphs. *arXiv preprint*, 2025.

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

[Yi *et al.*, 2023] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. FedPHP: Federated personalization with inherited private models. In *ECML-PKDD*, 2023.

[Yi *et al.*, 2024] Liping Yi, Gang Wang, Xiaoguang Liu, and Han Yu. FedSSA: Semantic similarity-based aggregation for efficient model-heterogeneous personalized federated learning. In *IJCAI*, 2024.

[Zhang *et al.*, 2021] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *ICLR*, 2021.

[Zhang *et al.*, 2022] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *NeurIPS*, 2022.

[Zhang *et al.*, 2023a] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. FedALA: Adaptive local aggregation for personalized federated learning. In *AAAI*, 2023.

[Zhang *et al.*, 2023b] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. FedCR: Personalized federated learning based on across-client common representation with conditional mutual information regularization. In *ICML*, 2023.

[Zhang *et al.*, 2023c] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. GPFL: Simultaneously learning global and personalized feature information for personalized federated learning. In *ICCV*, 2023.

[Zhang *et al.*, 2023d] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. PFLlib: A beginner-friendly and comprehensive personalized federated learning library and benchmark. *arXiv preprint arXiv:2312.04992*, 2023.

[Zhang *et al.*, 2023e] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Chen, and Yiran Wang. Towards building the federated GPT: Federated instruction tuning. *arXiv preprint arXiv:2305.05644*, 2023.

[Zhang *et al.*, 2024a] Jianqing Zhang, Yang Liu, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. FedTGP: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning. In *AAAI*, 2024.

[Zhang *et al.*, 2024b] Zhuo Zhang, Yuanhang Li, Yifan Ding, Xinyang Wang, Yang Liu, Yu Cheng, and Jinghui Zeng. FedBiOT: LLM local fine-tuning in federated learning without full model. In *KDD*, 2024.

[Zhou *et al.*, 2024] Yefan Zhou, Yujun Wang, Jinglong Ran, and Yaoqing Yang. Personalized federated fine-tuning for LLMs via data-driven heterogeneous model architectures. *arXiv preprint arXiv:2411.19128*, 2024.

[Zhu *et al.*, 2021] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, 2021.