Front. Comput. Sci., 2025, 0(0): 1–19
https://doi.org/10.1007/sxxxxx-yyy-zzzz-1

RESEARCH ARTICLE

FedAIMS: Adaptive Intermediate Supervision for Personalized Federated Learning

Shuyuan LI[1], Boyi LIU[1,2], Zimu ZHOU(✉)[1], Jin DONG(✉)[3]

1 Department of Data Science, City University of Hong Kong, Hong Kong 999077, China
2 State Key Laboratory of Complex & Critical Software Environment and Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China
3 Beijing Academy of Blockchain and Edge Computing, Beijing 100080, China

Abstract Personalized Federated Learning (PFL) enables the training of customized deep models on decentralized, heterogeneous data while preserving privacy. However, existing PFL methods primarily optimize the final layer, overlooking intermediate layers, which degrades backbone training, especially in non-IID settings. In this work, we propose FedAIMS (Federated Adaptive Intermediate Supervision), a novel PFL framework that incorporates intermediate supervision to enhance model training. FedAIMS adopts prototype-based feature alignment to provide effective intermediate supervision and adaptive supervision sampling to reduce computational overhead for resource-limited clients. Experiments on diverse datasets show that FedAIMS outperforms state-of-the-art PFL baselines by up to 36.76% in accuracy.

Keywords Personalized Federated Learning; Intermediate Supervision; Data Heterogeneity

1 Introduction

Personalized Federated Learning (PFL) [1] is an emerging paradigm for training customized deep models on decentralized, *heterogeneous* data. Unlike generic federated learning (GFL) [2], which learns a single global model, PFL tailors models for individual clients while keeping raw data localized. This makes it particularly valuable in privacy-sensitive, user-centric IoT applications, such as voice assistants [3], activity recognition [4], and health monitoring [5], intrusion detection [6] etc.

Among recent PFL advances, architecture partitioning strategies [7–12] have gained increasing

Received month dd, yyyy; accepted month dd, yyyy

E-mail: xxx@xxx.xxx

attention due to their compatibility with deep neural networks. These approaches divide models into a *global backbone* and *personalized heads*, allowing shared knowledge transfer while adapting to local data distributions. Yet existing methods focus on personalizing the *final* output layer, neglecting *intermediate* feature representations. This limitation leads personalized heads to overfit local data, which in turn *negatively impacts backbone training* [11, 13, 14], especially in non-IID settings.

To address this limitation, we introduce *intermediate supervision* to PFL, leveraging supervision signals from intermediate layers to improve backbone training and overall model performance. In centralized deep learning, deep supervision [15] enhances feature transparency by injecting auxiliary tasks at intermediate layers. However, applying intermediate supervision in PFL has two challenges.

- *Effective auxiliary tasks for PFL:* Unlike centralized training with large datasets, PFL clients operate on sparse, non-shareable data, limiting the use of traditional auxiliary tasks *e.g.* self-supervised classification [15, 16].
- *High computational costs for clients.* Many PFL clients are resource-constrained edge devices. Naively applying supervision to all intermediate layers significantly increases training overhead, making deployment infeasible.

In this paper, we propose FedAIMS, (Federated Adaptive InterMediate Supervision), a lightweight yet effective PFL framework with intermediate supervision. It is built upon two components. We introduce prototype-based feature alignment accompanied by personalized adapters for additional supervision for intermediate blocks. To reduce the training workload, we propose an adaptive supervision sampling strategy that reduces the local auxiliary task scale without compromising accuracy.

Our main contributions are:

- To the best of our knowledge, this is the first work that integrates intermediate supervision into personalized federated learning.
- We propose FedAIMS, a novel adaptive intermediate supervision framework for PFL, balancing supervision and efficiency through feature alignment and adaptive sampling.
- Experiments on diverse datasets show that FedAIMS outperforms PFL baselines [7–11, 17, 18] by up to 36.76% in accuracy.

2 Related Work

2.1 Personalized Federated Learning

Different from the traditional challenges of devices collaboration [19,20], Personalized Federated Learning (PFL) [1] addresses the limitations of Generic Federated Learning (GFL) [2,21–26], which struggles with low accuracy on heterogeneous client data. Instead of enforcing a single global model, PFL enables clients to collaboratively train models tailored to their local data. PFL approaches differ in their personalization strategies: *(i)* Architecture Partitioning. Split the model into global and personalized components [7–12], allowing partial parameter sharing while preserving client-specific representations. *(ii)* Model-wise Personalization. Apply regularization constraints [17,27] to balance local and global adaptation or use knowledge distillation [28, 29] to transfer knowledge without direct parameter sharing. *(iii)* Similarity-based Personalization. Leverage clustering [30, 31] or collaborative graphs [32, 33] to group clients based on inferred data similarities, refining models through selective knowledge exchange.

Among these strategies, architecture partitioning has gained significant attention due to its alignment with deep neural networks. By exploiting the layered structure of deep models, which naturally separate shared features from task-specific representations, it enables effective knowledge transfer while allowing fine-grained personalization. Representative methods include: FedRep [9], which selects a linear classifier as the personalized module and alternates between updating shared and personalized parts to ensure stable optimization; FedBABU [11], which freezes the classifier during federated training, and fine-tunes it locally to better capture client-specific representations; and mask-based approaches [34,35], which dynamically determine personalized sub-networks within a shared model by applying sparse masks.

While architecture partitioning based PFL leverages deep model modularity, existing solutions focus solely on optimizing the *final output layer*, overlooking hierarchical feature representations learned at intermediate layers. In contrast, we exploit *intermediate layers* as additional supervision signals to enhance model personalization.

2.2 Intermediate Supervision in Deep Learning

Intermediate supervision, or deep supervision [15], adds auxiliary objectives to the hidden layers [15] or side branches [16] of deep neural networks to improve training effectiveness. It has been used in various tasks such as image classification [15], face recognition [36], pose estimation [37], speech recognition [38], etc.

Originally proposed for self-improvement within a model, intermediate supervision has also gained popularity in knowledge distillation, where a large teacher model guides a smaller student model [39]. For instance, FitNets [40] aligns the hidden layer outputs of the student with those of the teacher, supervising the intermediate representations. Beyond feature activations, attention maps [41], probability distributions [42], and latent factors [43] are also commonly used for intermediate supervision.

Recently, intermediate supervision has been applied to federated learning. For example, FedIntR [44] introduces a contrastive loss at every block of the local models as representation regularization to improve accuracy. Our work introduces two key advancements. *(i)* We adapt intermediate supervision to personalized federated learning, addressing the challenge of heterogeneous data distributions across clients with *fine-grained* regularization on hidden layers. *(ii)* We propose *lightweight* yet effective intermediate supervision strategies, accounting for the communication cost of federated learning and resource constraints of local training.

3 Problem Statement

Scope. We consider personalized federated learning (PFL) [1] of deep neural networks with edge devices, as common in applications such as voice assistants [3], activity recognition [4], health monitoring [5], intrusion detection [6] etc. Assume n clients $\{c_1, \ldots, c_n\}$, with $\{D_1, \ldots, D_n\}$ as their local datasets, where the distribution of $\{D_i\}$ often varies across clients. PFL trains n personalized models $\{\theta_1, \ldots, \theta_n\}$ by optimizing the objective below:

$$\min \sum_{i=1}^{n} p_i F_i(\theta_i; D_i) \tag{1}$$

where $p_i = \frac{|D_i|}{\sum_{i=1}^{n} |D_i|}$, and F_i is the local objective of client i, for example, the cross-entropy loss of the model's output.

As mentioned in Sec. 2.1, we focus on *architecture partitioning* based PFL [7–12] because it utilizes the modularity in neural networks. Specifically, the model θ_i is partitioned into a global shared backbone ϕ_i and personalized head ψ_i, which is typically a linear classifier. Architecture partitioning based PFL relies on ϕ_i to maintain global knowledge while ψ_i to fit local distributions. During local training, both the global backbone ϕ_i and the personalized head ψ_i are updated at clients, but only the backbone ϕ_i is uploaded to the server for aggregation via *e.g.* FedAvg [2].

Problem. We aim to improve the accuracy of θ_i by introducing *auxiliary tasks* [15] at intermediate blocks of neural networks during PFL. Formally, we extend the objective in naive PFL *i.e.*, Eq. (1) to PFL with *intermediate supervision*:

$$\min \sum_{i=1}^{n} p_i \left(\alpha_m \underbrace{F_i(\theta_i; D_i)}_{\text{main task}} + \sum_{j=1}^{m-1} \alpha_j \underbrace{\mathcal{A}_i^j(\theta_i; D_i)}_{\text{auxiliary task}} \right) \quad (2)$$

where $F_i(\theta_i; D_i)$ is the main task for client i, which corresponds to the original local objective in naive PFL. $\mathcal{A}_i^j(\theta_i; D_i)$ for $1 \leq j \leq m-1$ is the auxiliary task added to block j of θ_i. α_j is a task-specific coefficient for block j, and $\sum_{j=1}^{m} \alpha_j = 1$ [45]. m is the total number of blocks in model θ_i.

The rationale to induce intermediate supervision in PFL is two-fold.

- As previously mentioned, intermediate supervision improves transparency of hidden layers within a single model. In PFL, the personalized heads tend to bias towards local datasets and negatively affect the training of the backbone [11, 13, 14]. Intermediate supervision allow the backbone to learn more discriminative and robust features [15], thus eliminating the bias from personalized heads.
- Intermediate supervision exploits the outputs of the hidden layers as regularization without extra cost, and it is compatible with architecture partitioning based PFL strategies.

Challenges. Implementing intermediate supervision in practical PFL systems faces two challenges.

- *How to design auxiliary tasks suited for federated learning?* Auxiliary tasks for self-improving of a single model [15, 16] underperform in federated learning due to the sparsity of local datasets. Intermediate supervision in knowledge distillation [40, 41] often requires sharing of raw data, which is prohibited in federated learning. Hence, we need new auxiliary tasks that are *cross-client* and *privacy-preserving*.
- *How to enable efficient training with intermediate supervision?* Adding supervision to all intermediate blocks as Eq. (2) may impose notable computation overhead to clients. Since clients in many practical PFL applications are resource-constrained devices, a *computational-efficient* intermediate supervision mechanism is desired.

4 Method

This section presents FedAIMS (Federated Adaptive InterMediate Supervision). Table. 1 lists the major notations that will be used throughout this paper.

4.1 FedAIMS Overview

Key Idea. FedAIMS enables *lightweight yet effective* intermediate supervision in PFL through *federated prototype alignment* and *adaptive supervision sampling*. Instead of enforcing supervision at all intermediate layers, FedAIMS employs prototype-based feature alignment to guide intermediate representations (see Sec. 4.2). To minimize computational overhead at clients, FedAIMS adaptively selects one intermediate block per client for supervision rather than involve all blocks (see Sec. 4.3).

Workflow. Each communication round in FedAIMS consists of three steps.

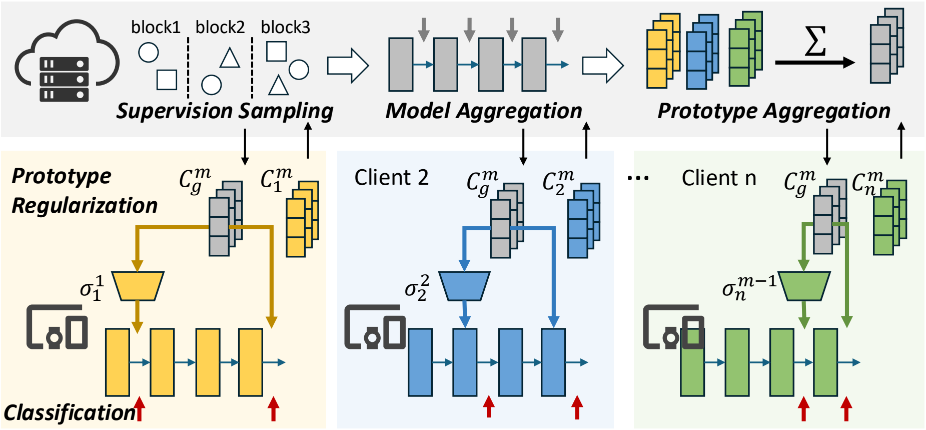


Fig. 1 FedAIMS overview. The process for the server mainly consists of supervision sampling (see Line 4 in Algorithm 1), model aggregation (see Line 15 in Algorithm 1), prototype aggregation (see Line 16 in Algorithm 1). The local objectives of clients consist of a classification task and prototype regularization.

Table 1 Summary of major notations.

Notation	Definition
n	number of clients
m	number of blocks
$D_i; \lvert D_i \rvert$	local dataset of client i; size of D_i
$D; \lvert D \rvert$	dataset of all clients; size of $\lvert D \rvert$
θ_i	model of client i, composed of ϕ_i and ψ_i
$\phi; \phi_i$	globally shared model backbone at server/client i
ψ_i	personalized head of client i
$\mathcal{A}_i^j$	auxiliary task of block j in client i
F_i	main task of client i
Φ_i	local objective of client i
Φ	global objective
E	epoch number
p_i	aggregation weight of client i
R_i^j	regularization term added to block j in client i
$C_{i,k}^j$	prototype of class k of block j in client i
$C_{g,k}^j$	global prototype of class k of block j
σ_i^j	adapter of block j at client i

- Step 1: *Sampling.* The server samples clients for training and clusters them into groups. Each cluster is assigned to supervise a specific intermediate block.
- Step 2: *Local Training.* Clients utilize the aggregated global prototype to supervise the sampled intermediate block via an adapter.
- Step 3: *Aggregation.* The server collects local models and prototypes from clients, performing weighted averaging to update both model parameters and prototypes [2, 46].

Algorithm 1 illustrates the detailed procedures of FedAIMS. In each communication round, the server first samples a subset of clients $\mathbb{C}$ (line 3). It then allocates a specific supervised block to each selected client (line 4). Afterwards, the server transmits both the model backbone ϕ^t and the global prototype C_g^m to the clients (lines 7-8). Each client first computes its local prototype C_i^m (line 9) and then updates its full model θ_i^t along with the corresponding adapter $\sigma_i^{b_i,t}$ (lines 11-12). Finally, the updated backbone and local prototype are uploaded to the server (line 13). All received backbones and prototypes are aggregated at the server (lines 15-16). The received backbones are also used to update the similarity matrix (line 17).

4.2 Federated Prototype Alignment

As discussed in Sec. 3, intermediate supervision in federated learning must be both cross-client and privacy-preserving. FedAIMS employs *prototypes*

Algorithm 1: FedAIMS

```
Input: Clients' model parameters $\theta_1, ..., \theta_n$
Output: Personalized models $\theta_1^T, ..., \theta_n^T$
for round $t$ do
    // sampling
    sample clients $\mathbb{C}$ for current round
    $\{b_i\} \leftarrow$ Supervision Sampling($\mathbb{C}$, $s$)
    // local training
    for client $i \in \mathbb{C}$ do
        receive $\phi^t, C_g^m$ from server
        replace local backbone $\phi_i^t \leftarrow \phi^t$
        calculate local prototype $C_i^m$ via
          Eq. (3)
        calculate overall training objective
          $\Phi_i$ via Eq. (12)
        update full model
          $\theta_i^{t+1} \leftarrow \theta_i^t - \eta \nabla_\theta \Phi_i(\theta_i^t; D_i)$
        update adapter
          $\sigma_i^{b_i,t+1} \leftarrow \sigma_i^{b_i,t} - \eta \nabla_\sigma \Phi_i(\theta_i^t; D_i)$
        upload $\phi_i^{t+1}, C_i^m$ to server
    // aggregation
    aggregate backbone $\phi^{t+1} \leftarrow \sum_{i \in \mathbb{C}} p_i \phi_i^{t+1}$
    aggregate prototype via Eq. (4)
    update similarity matrix $s$ with $\{\phi_i^t | i \in \mathbb{C}\}$
```

for feature-level supervision, which shares global knowledge across clients without exchanging raw data. While prototype-based alignment has been used in PFL to improve personalized head training [46, 47], we re-purpose it for intermediate supervision to enhance the global backbone.

- Each client computes per-class prototypes for its intermediate blocks, which are then aggregated at the server into a global prototype for intermediate supervision (Sec. 4.2.1).
- To reduce communication overhead, instead of transmitting prototypes for all intermediate blocks, clients only send the last block's prototype. An adapter then reconstructs block-wise prototypes during training (Sec. 4.2.2).

4.2.1 Block-Wise Prototypes to Align Features

Prototype Design. We define prototype $C_{i,k}^j$ as the averaged feature of block j for class k at client i:

$$C_{i,k}^j = \frac{\sum_{(x,y)\in D_i} \mathbf{1}_{y=k} \cdot \phi_i^j(x)}{\sum_{(x,y)\in D_i} \mathbf{1}_{y=k}} \tag{3}$$

where $(x, y) \in D_i$ represents a data sample and its label in client i's dataset, and $\phi_i^j(\cdot)$ is the output of block j in the backbone ϕ_i.

By telescoping the class-wise averaged features $C_{i,k}^j$ of all classes, we obtain the *local block-wise prototype* C_i^j for block j at client i.

The server then aggregates local block-wise prototypes $\{C_i^j\}_{i=1}^n$ from all clients into a *global block-wise prototype* C_g^j:

$$C_{g,k}^j = \sum_{i=1}^{n} p_i C_{i,k}^j \tag{4}$$

where is p_i is the aggregation weight for client i as defined in Eq. (1).

Intermediate Supervision via Prototypes. The global prototype C_g^j provides *feature alignment* at clients, serving as intermediate supervision. Given a data sample $(x, y) \in D_i$, we compute its intermediate feature $\phi_i^j(x)$ and regularize the distance between $\phi_i^j(x)$ and the corresponding prototype as:

$$R_i^j(\theta_i; D_i) = \sum_{(x,y)\in D_i} \left\| C_{g,y}^j - \phi_i^j(x) \right\|^2 \tag{5}$$

Finally, the main task is defined as a classification loss of the whole model and the prototype-based feature regularization:

$$F_i(\theta_i; D_i) = \mathcal{L}_{CE}^m(\theta_i; D_i) + \mu R_i^m(\theta_i; D_i) \tag{6}$$

The auxiliary task at block $j (1 \leq j \leq m-1)$ is the that of the intermediate block:

$$\mathcal{A}_i^j(\theta_i; D_i) = \mathcal{L}_{CE}^j(\theta_i; D_i) + \mu R_i^j(\theta_i; D_i) \tag{7}$$

where $\mathcal{L}_{CE}^j(\cdot)$ is the cross-entropy loss, and μ balances between classification and regularization.

4.2.2 Single-Block Prototypes to Reduce Communication Cost

While block-wise prototypes (Sec. 4.2.1) effectively align intermediate features, transmitting prototypes for all blocks introduces notable communication overhead. Hence, we only maintain the prototype of the *last block* and employ an *adapter* to recover per-block signals for intermediate supervision. The per-block adapter learns a projection function via *nonlinear* transformations, enabling the deep prototypes to be mapped into the shallow semantic feature space, thus facilitating better alignment [40].

Specifically, the server transmits only the global prototype of the last block C_g^m, and each client i uses an adapter σ_i^j to project C_g^m to match the dimensionality of features at block j. Consequently, given a data sample $(x, y) \in D_i$, the regularization of Eq. (5) is modified to align the mapped prototype $\sigma_i^j(C_{g,y}^m)$ with the shallow feature $\phi_i^j(x)$:

$$R_i^j(\theta_i; D_i) = \sum_{(x,y)\in D_i} \left\| \sigma_i^j(C_{g,y}^m) - \phi_i^j(x) \right\|^2 \tag{8}$$

In FedAIMS, the adapter σ_i^j is a personalized two-layer multilayer perceptron (MLP) maintained locally at each client. The model θ_i and adapter σ_i^j are jointly updated during local training using the objective in Eq. (7). Unlike prior methods [9, 10], our approach minimizes training overhead while ensuring effective intermediate supervision.

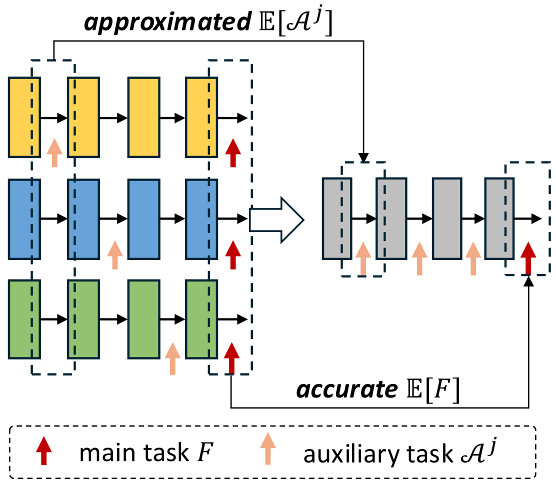

approximated $\mathbb{E}[\mathcal{A}^j]$
accurate $\mathbb{E}[F]$
main task F
auxiliary task $\mathcal{A}^j$

Fig. 2 Reformulating intermediate supervision.

4.3 Adaptive Supervision Sampling

Deriving supervision from all intermediate blocks across all clients in Eq. (2) incurs high computational overhead, making it impractical for resource-constrained edge devices [48, 49]. To address this, we propose an *adaptive supervision sampling* strategy that provides lightweight yet effective intermediate supervision. Specifically, we *sample only one* intermediate block from each client per round to minimize the overall computation cost, while ensuring effective intermediate supervision (see Fig. 2).

4.3.1 Reformulating Intermediate Supervision

We first explain the feasibility to *sample* the auxiliary tasks (*i.e.*, Eq. (2)) for effective supervision by reformulating them from client- to block-oriented. Given the assumption $\mathbb{E}[\mathcal{A}^j] = \sum_{i=1}^n p_i \mathcal{A}_i^j(\theta_i; D_i)$ [50], we rewrite the summation as:

$$\begin{aligned} \sum_{i=1}^{n} p_i \left(\sum_{j=1}^{m-1} \alpha_j \mathcal{A}_i^j(\theta_i; D_i) \right) &= \sum_{j=1}^{m-1} \alpha_j \sum_{i=1}^{n} p_i \mathcal{A}_i^j(\theta_i; D_i) \\ &= \sum_{j=1}^{m-1} \alpha_j \mathbb{E}[\mathcal{A}^j] \end{aligned} \tag{9}$$

This reformulation shows that auxiliary tasks can be viewed as a *block-wise expectation* over clients, allowing accurate estimation of $\mathbb{E}[\mathcal{A}^j]$ without requiring participation from all clients.

We make two observations.

- *Redundant clients can be omitted.* Clients with similar data distributions tend to generate similar auxiliary tasks, since $\mathcal{A}_i^j(\theta_i; D_i)$ depends on D_i.
- *Shallower blocks require fewer clients.* Features at shallower blocks are more universal and similar across clients [51], reducing the number of clients needed to estimate $\mathbb{E}[\mathcal{A}^j]$ at a shallower block j.

4.3.2 Supervision Sampling Strategy

Principles. Based on the observations in Sec. 4.3.1, we develop an *adaptive* sampling strategy to approximate $\mathbb{E}[\mathcal{A}^j]$, guided by two principles. *(i)* Assign more clients to deeper blocks to improve estimation accuracy where feature representations are more specialized and personalized. *(ii)* Ensure unbiased block-wise expectation $\mathbb{E}[\mathcal{A}^j]$, avoiding over-representation of any particular data distribution.

Clustering-Based Sampling. To satisfy these principles, we assign clients and blocks for estimating $\mathbb{E}[\mathcal{A}^j]$ through a clustering-based strategy.

We cluster sampled clients into $m-1$ groups $\{G_1, \ldots, G_{m-1}\}$, where each group serves as supervision for a specific intermediate block.

The clustering objective is formulated as:

$$\min \sum_{k=1}^{m-1} \sum_{i,j \in G_k} \cos(\phi_i, \phi_j) \qquad (10)$$
$$s.t. \;\; |G_i| - |G_j| \leq 1 \; \forall 1 \leq i, j \leq m-1$$

where $\cos(\phi_i, \phi_j)$ measures similarity between client models. This ensures:

- *cosine similarity* as the clustering metric, commonly used in PFL [30, 31].
- *uniform* group sizes, with a tendency to allocate *more* clients to *deeper* blocks.
- *minimal* intra-group similarity, preventing bias in estimating $\mathbb{E}[\mathcal{A}^j]$.

Note that the similarity calculation and collaboration is widely applied in similarity-based PFL [30, 31]. FedAIMS does not introduce any additional computational overhead for similarity calculation compared to these existing methods.

Sampling Algorithm. To iteratively form groups, we define the additional similarity contribution $\delta_{i,k}$ from adding client i to group G_k as:

$$\delta_{i,k} = \frac{\sum_{j \in G_k} \cos(\phi_i, \phi_j)}{|G_k|} \qquad (11)$$

Clients are added to the group that minimizes $\delta_{i,k}$, ensuring diverse assignments. To balance group sizes, we always assign clients to the smallest group first. Groups are then sorted by size, with larger groups assigned to deeper blocks, ensuring more supervision where needed. The full process is detailed in Algorithm 2.

Approximated Auxiliary Tasks. Once clients are assigned to supervise specific blocks, the training objective for client i in Eq. (2) simplifies to:

$$\min \; \Phi_i = \lambda F_i(\theta_i; D_i) + (1-\lambda)\mathcal{A}_i^j(\theta_i; D_i) \qquad (12)$$

where $\mathcal{A}_i^j$ follows Eq. (7). We set $\lambda = 1/m$ to ensure uniform weighting of supervision across all blocks after aggregation.

4.4 Analysis and Discussion

4.4.1 Convergence

Under assumptions in Appendix A, our proposed FedAIMS algorithm converges (see Theorem 1). We extend the analysis in [52] from a traditional

Algorithm 2: Supervision Sampling

```
Input: Selected clients $\mathbb{C}$, Similarities $s$
Output: Allocation index $\{b_i | i \in \mathbb{C}\}$
// clustering
initialize $G_1, \dots, G_{m-1}$ with $\emptyset$
for client $i \in \mathbb{C}$ do
    group index $k \leftarrow \arg\min \delta_{i,k}$ and satisfy
      constraint in Eq. (10)
    $G_k \leftarrow G_k \cup \{i\}$
// sampling
sort groups with scale in ascending order
for block $k \leftarrow m-1$ to 1 do
    for client $i \in G_k$ do
        block index $b_i \leftarrow k$
```

PFL strategy to PFL with intermediate supervision. All proofs are in Appendix A.

Theorem 1. (Convergence of FedAIMS). FedAIMS converges to an arbitrary constant ϵ with a convergence rate of $O(1/T)$ when the learning rate satisfies $\eta < \min\left(\frac{2(\epsilon-\kappa^2)}{L(\epsilon+E\sigma^2)}, \frac{2}{L}\right)$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\left(\lambda\left\|\nabla F_i^{tE+e}\right\|^2 + (1-\lambda)\left\|\nabla \mathcal{A}_i^{tE+e}\right\|^2\right) \leq \frac{\frac{\Delta}{T} + \frac{LE\eta^2\sigma^2}{2} + \eta\kappa^2}{\eta - \frac{L\eta^2}{2}} < \epsilon \tag{13}$$

where $\Delta = \Phi_i^{(T-1)E+0} - \Phi_i^{0E+0}$, T is the communication round, E is the local epoch, η is the learning rate, and L, σ, κ are explained in assumptions.

4.4.2 Communication Cost

We compare the communication cost of FedAIMS with standard GFL [2,21–26] and representative architecture partitioning based PFL [7–12].

- Standard GFL transmits the entire model, including the global *backbone* and model *head*.
- Architecture partitioning based PFL reduces communication cost by transmitting only a *partial* model *e.g.* backbone only.
- FedAIMS transmits a *partial* model along with *prototypes*, but we show that it does not induce additional overhead than standard GFL.

Let D be the hidden dimension of the final block, and C the number of classes. Since a prototype for each class has dimension D, the total size of the prototype matrix is $D * C$. Standard GFL must transmit the model head, whose size is also $D * C$. Thus, FedAIMS and standard GFL have an equivalent communication cost, as FedAIMS replaces model head transmission with prototype transmission. We further provide an empirical study to support this conclusion (see Sec. 5.3.6).

4.4.3 Privacy

FedAIMS uploads both the *local prototype* and the *model backbone* to the server (see line 13 in Algorithm 1). As pointed out in [46], prototypes, which are computed as the mean embeddings of samples from the same class, are inherently irreversible, thus providing natural data privacy protection. Moreover, the model backbone represents only a sub-model of the full local model. In conclusion, sharing the backbone in FedAIMS provides at least comparable privacy protection to most existing FL baselines. The privacy protection is not the main focus of FedAIMS, and extra security techniques can also be applied to the prototypes for further privacy enhancement.

5 Experiment

5.1 Experimental Setup

Environment. The experiments are conducted on a machine equipped with an Intel Xeon Gold 6230R

Table 2 *Accuracy* over four datasets under $\eta = 0.1$.

Method	EMNIST		CIFAR-10		CIFAR-100		TinyImageNet	
	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)
Local	$77.98_{\pm 7.76}$	$87.72_{\pm 8.08}$	$66.69_{\pm 13.15}$	$80.49_{\pm 16.30}$	$23.48_{\pm 5.25}$	$41.04_{\pm 10.39}$	$19.10_{\pm 13.40}$	$32.47_{\pm 15.75}$
FedAvg [2]	$79.40_{\pm 6.48}$	$79.51_{\pm 9.96}$	$26.26_{\pm 16.48}$	$14.00_{\pm 28.08}$	$31.54_{\pm 5.28}$	$20.19_{\pm 6.76}$	$19.64_{\pm 4.90}$	$11.01_{\pm 4.43}$
FedProx [21]	$81.64_{\pm 5.58}$	$79.31_{\pm 9.66}$	$62.47_{\pm 13.43}$	$56.85_{\pm 14.73}$	$30.26_{\pm 5.10}$	$22.62_{\pm 6.74}$	$19.79_{\pm 4.94}$	$12.74_{\pm 4.92}$
MOON [22]	$80.44_{\pm 6.43}$	$80.20_{\pm 10.43}$	$52.24_{\pm 12.60}$	$10.27_{\pm 22.64}$	$32.79_{\pm 5.58}$	$17.64_{\pm 6.63}$	$21.67_{\pm 5.59}$	$13.33_{\pm 6.23}$
FedPer [7]	$85.14_{\pm 4.50}$	$91.80_{\pm 3.89}$	$75.40_{\pm 14.50}$	$86.22_{\pm 10.80}$	$32.70_{\pm 5.29}$	$55.22_{\pm 9.23}$	$26.66_{\pm 12.07}$	$44.93_{\pm 13.66}$
FedRep [9]	$83.94_{\pm 5.23}$	$91.21_{\pm 4.28}$	$74.23_{\pm 11.37}$	$86.36_{\pm 10.86}$	$30.09_{\pm 5.85}$	$48.57_{\pm 9.45}$	$23.34_{\pm 12.62}$	$41.32_{\pm 14.08}$
LG-FedAvg [8]	$76.46_{\pm 8.19}$	$87.21_{\pm 5.49}$	$66.70_{\pm 12.82}$	$8.91_{\pm 23.33}$	$23.87_{\pm 5.88}$	$40.73_{\pm 10.38}$	$20.14_{\pm 13.18}$	$33.62_{\pm 15.43}$
Ditto [17]	$78.23_{\pm 5.22}$	$90.88_{\pm 7.63}$	$67.47_{\pm 13.08}$	$80.71_{\pm 15.80}$	$23.75_{\pm 5.37}$	$41.81_{\pm 10.24}$	$19.81_{\pm 13.32}$	$33.66_{\pm 15.53}$
FedBABU [11]	$80.78_{\pm 4.83}$	$90.79_{\pm 4.31}$	$73.27_{\pm 11.65}$	$87.12_{\pm 10.53}$	$29.38_{\pm 8.12}$	$49.93_{\pm 14.42}$	$27.54_{\pm 14.08}$	$20.00_{\pm 23.89}$
FedRoD [10]	$76.34_{\pm 5.53}$	$88.63_{\pm 4.72}$	$76.87_{\pm 10.27}$	$87.81_{\pm 10.05}$	$35.95_{\pm 6.48}$	$53.73_{\pm 8.44}$	$28.65_{\pm 12.55}$	$42.82_{\pm 13.64}$
FedALA [18]	$81.95_{\pm 4.69}$	$83.63_{\pm 10.62}$	$60.16_{\pm 14.89}$	$66.19_{\pm 17.45}$	$32.79_{\pm 5.49}$	$21.50_{\pm 6.19}$	$19.37_{\pm 4.20}$	$14.78_{\pm 11.92}$
FedAIMS	$\mathbf{88.85}_{\pm 3.54}$	$\mathbf{93.33}_{\pm 3.46}$	$\mathbf{81.79}_{\pm 8.04}$	$\mathbf{89.45}_{\pm 8.72}$	$\mathbf{48.52}_{\pm 6.01}$	$\mathbf{58.26}_{\pm 8.86}$	$\mathbf{33.17}_{\pm 11.71}$	$\mathbf{46.43}_{\pm 13.65}$

CPU and NVIDIA A100 GPUs (40GB memory).

Baselines. We compare FedAIMS with the following representative FL baselines:

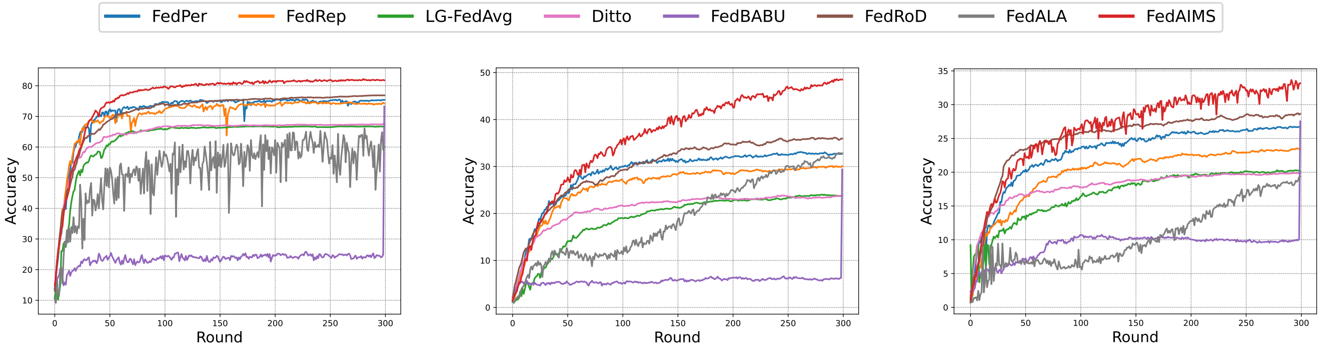
- **Local**: Each client trains its model independently with its local dataset.
- **FedAvg** [2]: Average model parameters to train a global model.
- **FedProx** [21]: Extend FedAvg with an additional regularization term to enforce similarity between local and global models.
- **MOON** [22]: Incorporate a model-contrastive loss into FedAvg.
- **FedPer** [7]: Aggregate only the backbone while jointly updating the personalized head and global backbone.
- **FedRep** [9]: Aggregate only the backbone but update the personalized head and global backbone iteratively.
- **LG-FedAvg** [8]: Aggregate only the head for personalized model training.
- **Ditto** [17]: Train a global and multiple personalized models, using the global model for regularization.
- **FedBABU** [11]: Freeze the backbone during training and fine-tune the personalized head at test time.
- **FedRoD** [10]: Introduce an additional personalized head, combining outputs from the global and personalized heads.
- **FedALA** [18]: Adaptively aggregate the global model at clients as personalized models.

Datasets and Models. We evaluate FedAIMS on four datasets: EMNIST [53], CIFAR-10 [54], CIFAR-100 [54], and TinyImageNet [55]. For EMNIST, we use the *Balanced* subset containing 47 classes. To simulate data heterogeneity, we follow [34] and apply a Dirichlet distribution [56] with two heterogeneity levels: Dir(0.3) and Dir(0.1). Following [11], we use a 3-layer ConvNet for EMNIST and a 4-layer ConvNet for CIFAR-10. For CIFAR-100 and TinyImageNet, we use ResNet-18 [57].

Configurations. We simulate 100 clients and a client sampling rate of 0.1. The number of communication rounds T is 300, with $E = 5$ local epochs per round. The batch size is 64, and the learning rate η is 0.1. We use SGD as the optimizer with a

Table 3 *Accuracy* over four datasets under $\eta = 0.05$.

Method	EMNIST		CIFAR-10		CIFAR-100		TinyImageNet	
	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)	Dir(0.3)	Dir(0.1)
Local	$78.77_{\pm 4.87}$	$88.93_{\pm 4.67}$	$67.73_{\pm 12.61}$	$82.36_{\pm 13.10}$	$25.15_{\pm 5.65}$	$43.68_{\pm 9.69}$	$20.59_{\pm 13.12}$	$34.43_{\pm 15.43}$
FedAvg [2]	$82.02_{\pm 5.07}$	$81.06_{\pm 7.86}$	$66.10_{\pm 7.40}$	$34.74_{\pm 20.39}$	$29.80_{\pm 4.87}$	$18.58_{\pm 5.64}$	$15.71_{\pm 4.82}$	$10.07_{\pm 5.41}$
FedProx [21]	$81.12_{\pm 5.38}$	$81.12_{\pm 8.00}$	$66.21_{\pm 6.96}$	$58.64_{\pm 16.19}$	$29.19_{\pm 5.15}$	$20.22_{\pm 7.08}$	$12.68_{\pm 4.18}$	$10.32_{\pm 4.74}$
MOON [22]	$80.96_{\pm 5.60}$	$79.15_{\pm 10.47}$	$62.95_{\pm 10.07}$	$45.81_{\pm 20.79}$	$31.36_{\pm 4.76}$	$23.07_{\pm 6.61}$	$16.73_{\pm 5.39}$	$11.58_{\pm 4.47}$
FedPer [7]	$85.79_{\pm 4.46}$	$92.52_{\pm 3.37}$	$76.91_{\pm 10.09}$	$87.88_{\pm 9.18}$	$37.97_{\pm 6.45}$	$55.52_{\pm 8.81}$	$29.04_{\pm 11.61}$	$45.10_{\pm 12.86}$
FedRep [9]	$83.18_{\pm 4.66}$	$91.00_{\pm 4.03}$	$73.12_{\pm 10.97}$	$86.39_{\pm 10.42}$	$33.89_{\pm 5.48}$	$52.78_{\pm 9.23}$	$25.13_{\pm 12.19}$	$44.73_{\pm 13.31}$
LG-FedAvg [8]	$78.49_{\pm 5.01}$	$88.53_{\pm 5.13}$	$68.09_{\pm 13.02}$	$82.76_{\pm 13.28}$	$25.34_{\pm 5.55}$	$43.06_{\pm 9.37}$	$21.73_{\pm 12.75}$	$35.16_{\pm 15.45}$
Ditto [17]	$78.90_{\pm 5.41}$	$89.73_{\pm 5.21}$	$67.52_{\pm 13.46}$	$82.77_{\pm 12.82}$	$25.55_{\pm 5.57}$	$43.76_{\pm 10.04}$	$20.88_{\pm 13.11}$	$34.84_{\pm 15.35}$
FedBABU [11]	$79.47_{\pm 5.27}$	$90.37_{\pm 4.37}$	$74.42_{\pm 11.01}$	$87.10_{\pm 10.54}$	$26.21_{\pm 9.68}$	$45.70_{\pm 16.26}$	$22.81_{\pm 13.60}$	$37.32_{\pm 18.24}$
FedRoD [10]	$72.26_{\pm 7.33}$	$87.55_{\pm 5.10}$	$76.89_{\pm 9.93}$	$88.21_{\pm 8.83}$	$35.24_{\pm 7.45}$	$52.28_{\pm 8.53}$	$26.64_{\pm 12.93}$	$40.00_{\pm 14.33}$
FedALA [18]	$81.73_{\pm 4.59}$	$83.86_{\pm 5.78}$	$66.18_{\pm 6.57}$	$67.48_{\pm 12.16}$	$30.24_{\pm 5.68}$	$19.68_{\pm 6.38}$	$16.53_{\pm 4.46}$	$13.32_{\pm 10.88}$
FedAIMS	$\mathbf{88.64}_{\pm 3.56}$	$\mathbf{93.12}_{\pm 3.49}$	$\mathbf{82.37}_{\pm 8.28}$	$\mathbf{89.01}_{\pm 8.95}$	$\mathbf{45.32}_{\pm 6.92}$	$\mathbf{55.70}_{\pm 9.43}$	$\mathbf{31.29}_{\pm 12.37}$	$\mathbf{45.21}_{\pm 12.72}$



(a) CIFAR-10 **(b)** CIFAR-100 **(c)** TinyImageNet

Fig. 3 Accuracy curve of PFL baselines in Dir(0.3).

momentum of 0.9, weight decay of 1×10^{-4}, and a learning rate decay of 0.99.

Hyperparameters for the methods are as follows:

- FedProx: $\mu = 0.5$.
- MOON: $\tau = 0.5, \mu = 1$.
- FedRep: Personalized head training epochs $E_p = 5$.
- Ditto: $E_p = 5, \lambda = 0.1$.
- FedBABU: Fine-tuning epochs $E_{ft} = 5$.
- FedRoD: $E_p = 5$.
- FedALA: Random sampling ratio = 0.1, local aggregation learning rate $\eta_l = 0.1$.
- FedAIMS: $\mu = 1$.

5.2 Main Results

Table. 2 reports the accuracy of FedAIMS and baselines. Under the non-IID scenario, PFL baselines consistently outperform GFL baselines. Compared to the PFL baselines, FedAIMS achieves higher accuracy across two levels of data heterogeneity. Specifically, the improvements are as follows: up to 12.51%, 9.70% on EMNIST, 21.63%, 23.26% on CIFAR-10, 24.77%, 36.76% on CIFAR-100, and 14.76%, 31.89% on TinyImageNet.

Fig. 3 shows the accuracy-round curves, where only PFL baselines are included for comparison. FedAIMS not only converges faster but also achieves higher accuracy than all PFL baselines. Notably, its advantage is more pronounced in challenging datasets such as CIFAR-100 and TinyImageNet. As shown in Fig. 3, although the maximum communication rounds is $T = 300$, FedAIMS could achieve even higher accuracy with additional rounds. In contrast, other baselines plateau at a relatively steady accuracy by this point.

5.3 Ablation Study

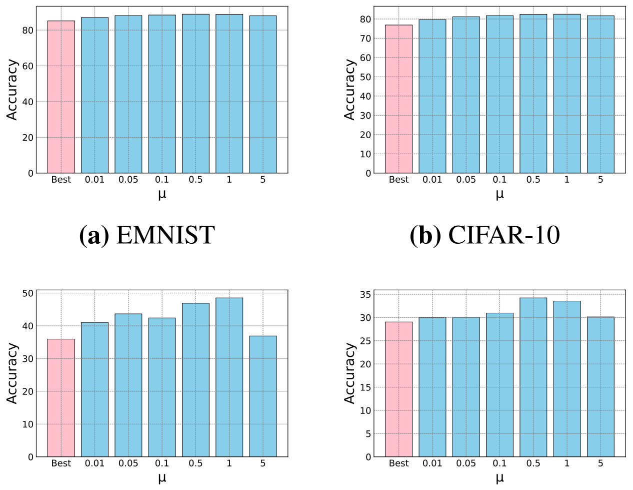
5.3.1 Impact of Hyperparameter

We tune the hyperparameter μ, which balances classification and prototype regularization, over the range $0.01, 0.05, 0.1, 0.5, 1, 5$. Fig. 4 shows the results.

For EMNIST and CIFAR-10, FedAIMS remains robust across different values of μ and consistently outperforms the best PFL baselines. For more challenging datasets, CIFAR-100 and TinyImageNet, FedAIMS is more sensitive to μ due to increased task complexity and model size. However, it still surpasses all PFL baselines when $\mu \in [0.01, 1]$. Empirically, we find that FedAIMS performs well when $\mu = 0.5$ or $\mu = 1$ and fix $\mu = 1$ for all datasets. We will explore task-adaptive configuration of μ to further improve FedAIMS in future work.

5.3.2 Impact of Learning Rate

We evaluate FedAIMS and baseline methods under a reduced learning rate of $\eta = 0.05$. The results are in Table. 3. FedAIMS consistently outperforms all PFL baselines in this setting. The improvements are as follows: up tp 16.38%, 9.36% in EMNIST, 16.19%, 22.53% in CIFAR-10, 19.98%, 36.02% in CIFAR-100, 14.76%, 41.89% in TinyImageNet.



(a) EMNIST

(b) CIFAR-10

(c) CIFAR-100

(d) TinyImageNet

Fig. 4 Hyperparameter analysis on four datasets. *Best* refers to the best performance reported in PFL baselines.

5.3.3 Impact of Other Data Heterogeneity

We evaluate FedAIMS under the pathological distribution [2], where each client holds only C classes. Following [18], we set $C = 10$ for EMNIST, $C = 2$ for CIFAR-10, and $C = 10$ for CIFAR-100. Table. 4 shows the results. FedAIMS outperforms all PFL baselines, improving accuracy by 0.46-12.14% on EMNIST, 1.97-40.02% on CIFAR-10, and 4.95-39.49% on CIFAR-100.

5.3.4 Impact of Similarity Matrix Update Frequency

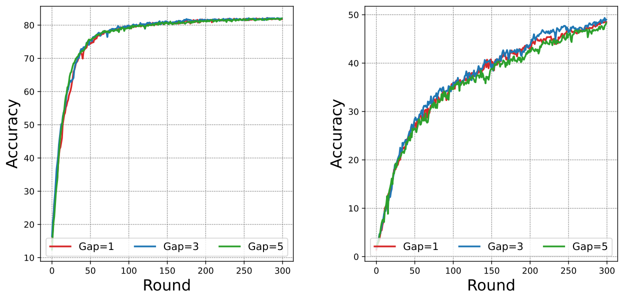
Fig. 5 shows the impact of similarity matrix update frequency on the convergence speed of FedAIMS. In FedAIMS, the similarity matrix is updated every round by default. We tune the update round gap in $[1, 3, 5]$. As is shown, the convergence speed is insensitive to the similarity matrix update frequency.

5.3.5 Comparison with Full Supervision

Table. 5 compares FedAIMS with its fully supervised variant, where supervision is applied to all

Table 4 *Accuracy* over pathological distribution.

Method	EMNIST	CIFAR-10	CIFAR-100
	#C=10	#C=2	#C=10
Local	$88.80_{\pm 8.60}$	$61.38_{\pm 23.94}$	$36.07_{\pm 6.49}$
FedAvg [2]	$78.61_{\pm 12.79}$	$10.07_{\pm 20.19}$	$53.85_{\pm 9.42}$
FedProx [21]	$79.08_{\pm 13.69}$	$9.93_{\pm 19.92}$	$65.96_{\pm 8.35}$
MOON [22]	$76.22_{\pm 13.04}$	$9.98_{\pm 20.00}$	$47.75_{\pm 10.17}$
FedPer [7]	$93.26_{\pm 2.90}$	$83.69_{\pm 10.13}$	$64.09_{\pm 5.67}$
FedRep [9]	$93.08_{\pm 2.97}$	$87.64_{\pm 6.30}$	$60.35_{\pm 5.76}$
LG-FedAvg [8]	$89.54_{\pm 3.58}$	$61.43_{\pm 22.85}$	$29.51_{\pm 8.38}$
Ditto [17]	$92.55_{\pm 3.82}$	$60.37_{\pm 21.25}$	$37.50_{\pm 7.16}$
FedBABU [11]	$93.15_{\pm 3.16}$	$86.67_{\pm 6.92}$	$60.01_{\pm 13.04}$
FedRoD [10]	$93.71_{\pm 2.66}$	$87.85_{\pm 5.20}$	$62.45_{\pm 5.78}$
FedALA [18]	$82.03_{\pm 9.42}$	$49.59_{\pm 3.72}$	$51.89_{\pm 9.57}$
FedAIMS	$\mathbf{94.17}_{\pm 3.09}$	$\mathbf{89.61}_{\pm 5.64}$	$\mathbf{69.02}_{\pm 5.67}$



(a) CIFAR-10 **(b)** CIFAR-100

Fig. 5 Impact of similarity matrix update frequency.

intermediate blocks. We evaluate both methods under two non-IID settings. FedAIMS achieves performance comparable to the fully supervised variant, with accuracy differences within 1% in most cases. This highlights the effectiveness of adaptive supervision sampling, allowing FedAIMS to maintain similar accuracy while significantly reducing computational overhead.

5.3.6 Communication Cost

Table 5 *Accuracy* of FedAIMS and its variant with full supervision (denoted as FedAIMS-Full).

Dataset		FedAIMS	FedAIMS-Full
EMNIST	Dir(0.3)	88.85	88.38(**−0.47**)
	Dir(0.1)	93.33	93.26(**−0.07**)
CIFAR-10	Dir(0.3)	81.79	82.65(**+0.86**)
	Dir(0.1)	89.45	89.34(**−0.11**)
CIFAR-100	Dir(0.3)	48.52	48.96(**+0.44**)
	Dir(0.1)	58.26	55.93(**−2.33**)
TinyImageNet	Dir(0.3)	33.17	32.11(**−1.06**)
	Dir(0.1)	46.43	45.77(**−0.66**)

Table. 6 presents the theoretical and empirical communication cost of FedAIMS and the baselines. Since all methods have the same training rounds of T = 300 (see Sec. 5.1), we report the *averaged communication cost per round* in both the theoretical and empirical analyses. Let Σ be the model size, β the proportion of the shared backbone, and π the size of the prototype.

From a theoretical perspective, most FL baselines transmit the entire model, incurring a communication cost of $2 * \Sigma$ per round. Architecture partitioning based PFL methods, which transmit only a portion of the model, reduce this cost to $2*\beta*\Sigma$. For FedAIMS, the communication cost is $2 * \beta * \Sigma + \pi$.

We further validate this with practical results on four datasets, showing that FedAIMS's actual communication cost $2 * \beta * \Sigma + \pi$ is comparable to the $2 * \Sigma$ cost of most FL baselines. This confirms the theoretical analysis in Sec. 4.4.2, demonstrating that FedAIMS does not introduce notable extra communication overhead.

5.3.7 Training Time

Table. 7 shows the average training time per round for FedAIMS (our proposed method with sampling), and FedAIMS-Full (variant that applies supervision to all intermediate blocks without sampling).

Table 6 Empirical and theoretical communication cost.

Method	Practical				Theoretical
	EMNIST	CIFAR-10	CIFAR-100	TinyImageNet	
Model Size	0.30 MB	1.71 MB	42.80 MB	43.00 MB	Σ
Local	0 MB	0 MB	0 MB	0 MB	0
FedAvg [2]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
FedProx [21]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
MOON [22]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
FedPer [7]	0.57 MB	3.41 MB	85.21 MB	85.21 MB	$2 * \beta * \Sigma$
FedRep [9]	0.57 MB	3.41 MB	85.21 MB	85.21 MB	$2 * \beta * \Sigma$
LG-FedAvg [8]	0.02 MB	0.01 MB	0.39 MB	0.78 MB	$2 * (1 - \beta) * \Sigma$
Ditto [17]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
FedBABU [11]	0.57 MB	3.41 MB	85.21 MB	85.21 MB	$2 * \beta * \Sigma$
FedRoD [10]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
FedALA [18]	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * \Sigma$
FedAIMS	0.59 MB	3.42 MB	85.60 MB	85.99 MB	$2 * (\beta * \Sigma + \pi)$

Compared to FedAIMS-Full, which supervises all intermediate layers, FedAIMS reduces training time by 14.64%, 12.59%, 12.61% and 9.71% across different settings without or with mild loss in accuracy (see Table. 5). The reduced training time validates the effectiveness of the adaptive supervision sampling.

Table 7 *Averaged training time per round* of FedAIMS and its variant without adaptive sampling (denoted as FedAIMS-Full).

Dataset	FedAIMS	FedAIMS-Full
EMNIST	4.08s	4.78s
CIFAR-10	4.86s	5.56s
CIFAR-100	6.03s	6.90s
TinyImageNet	10.31s	11.42s

5.3.8 Communication-Storage Tradeoff

The adapter in FedAIMS maps the prototype of the final block to shallow prototypes, which reduces the communication cost associated with prototype transmission at the expense of additional storage. Table. 8 compares the additional storage overhead introduced by the adapter and the corresponding reduction in communication cost during federated training. Overall, the adapter substantially lowers the total communication cost while incurring minimal extra storage overhead.

Table 8 Tradeoff between storage cost and communication cost due to adapters.

Dataset	Storage	Communication
EMNIST	+0.06MB	-13.77MB
CIFAR-10	+0.19MB	-8.79MB
CIFAR-100	+0.49MB	-219.73MB
TinyImageNet	+0.49MB	-439.45MB

6 Conclusion

This paper presents FedAIMS, a novel PFL framework with adaptive intermediate supervision. With prototype-based feature alignment and selective supervision sampling, FedAIMS enhances model personalization while maintaining low computational and communication costs. Our work highlights the potential of intermediate supervision in improving PFL and provides a lightweight solution for practical PFL applications.

Acknowledgements This work was partially supported by National Science Foundation of China (NSFC) (Grant Nos. 62425202, U21A20516, 62336003), the CityU APRC grant (No. 9610633), the Beijing Natural Science Foundation (Z230001), the Fundamental Research Funds for the Central Universities No. JK2024-03, the Didi Collaborative Research Program and the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). Zimu Zhou and Jin Dong are the corresponding authors.

Appendixes

Appendix A Proof of Theorem 1

In this section, we present the proof of Theorem 1 to support the convergence analysis of FedAIMS. Specifically, we extend the analysis framework in [52] by incorporating intermediate supervision.

Preliminary. We first define the basic notations for convergence analysis. There are a total of T communication rounds, and each round consists of E epochs of local training. $tE + 0$ denotes the start of the t-th communication round, where clients receive the global model. $tE+e$ denotes the e-th local epoch within the t-th communication round. After completing local training, clients upload the shared backbone ϕ for aggregation at the end of the t-th round, which corresponds to $tE + E$.

For ease of presentation, we denote the main task F_i as F_i^m, and the auxiliary task $\mathcal{A}_i^j$ as F_i^j. We then make the following assumptions:

Assumption 1. (Smoothness). The objective F_i^j at block j of client i is L-smooth, satisfying:

$$\left\|\nabla F_i^j(x) - \nabla F_i^j(y)\right\| \leq L\left\|x - y\right\| \tag{14}$$

which can be further reformulated as:

$$F_i^j(y) \leq F_i^j(x) + \langle \nabla F_i^j(x), y - x\rangle + \frac{L}{2}\left\|y - x\right\|^2 \tag{15}$$

Assumption 2. (Unbiased gradient estimator). The expectation of stochastic gradient $\nabla F_i^j(\theta_i;\xi_i)$ is an unbiased estimator of the local gradient, satisfying:

$$\mathbb{E}_{\xi_i\sim D_i}\nabla F_i^j(\theta_i;\xi_i) = \nabla F_i^j(\theta_i) \tag{16}$$

Assumption 3. (Bounded gradient variance). The variance of stochastic gradient $\nabla F_i^j(\theta_i;\xi_i)$ is bounded by σ^2, satisfying:

$$\mathbb{E}_{\xi_i\sim D_i}\left\|\nabla F_i^j(\theta_i;\xi_i) - \nabla F_i^j(\theta_i)\right\|^2 \leq \sigma^2 \tag{17}$$

Assumption 4. (Bounded backbone variation). The parameter variations of the shared backbone ϕ_i^t and ϕ^{t+1} before and after aggregation at server are bounded, satisfying:

$$\left\|\phi^{t+1} - \phi_i^t\right\|^2 \leq \kappa^2 \tag{18}$$

Lemma 1. (Bounding local training). Given Assumption 1, Assumption 2 and Assumption 3, the loss of an arbitrary client i at an arbitrary communication round t is bounded by:

$$\begin{aligned}\mathbb{E}\left[F_i^{tE+E}\right] \leq& F_i^{tE+0} + \frac{LE\eta^2\sigma^2}{2} \\ &+ \left(\frac{L\eta^2}{2} - \eta\right)\sum_{e=0}^{E-1}\left\|\nabla F_i^{tE+e}\right\|^2\end{aligned} \tag{19}$$

Proof. We first analyze the change of loss of a single block j in a single epoch. Take round $tE + 0$ as an example, based on Assumption 1, we have:

$$\begin{aligned}F_i^{j,tE+1} \leq& F_i^{j,tE+0} - \left\langle \nabla F_i^{j,tE+0}, \theta_i^{tE+1} - \theta_i^{tE+0}\right\rangle \\ &+ \frac{L}{2}\left\|\theta_i^{tE+1} - \theta_i^{tE+0}\right\|^2 \\ =& F_i^{j,tE+0} - \eta\left\langle \nabla F_i^{j,tE+0}, g_i^{j,tE+0}\right\rangle + \frac{L\eta^2}{2}\left\|g_i^{j,tE+0}\right\|^2\end{aligned} \tag{20}$$

We take expectation on both sides, and get:

$$
\begin{aligned}
\mathbb{E}\left[F_i^{j,tE+1}\right] \le & F_i^{j,tE+0} - \eta\mathbb{E}\left[\left\langle \nabla F_i^{j,tE+0}, g_i^{j,tE+0}\right\rangle\right] \\
& + \frac{L\eta^2}{2}\mathbb{E}\left[\left\|g_i^{j,tE+0}\right\|^2\right] \\
\le & F_i^{j,tE+0} - \eta\left\|\nabla F_i^{j,tE+0}\right\|^2 \\
& + \frac{L\eta^2}{2}\left(\left\|\nabla F_i^{j,tE+0}\right\|^2 + Var(g_i^{j,tE+0})\right) \\
\le & F_i^{j,tE+0} - \eta\left\|\nabla F_i^{j,tE+0}\right\|^2 \\
& + \frac{L\eta^2}{2}\left(\left\|\nabla F_i^{j,tE+0}\right\|^2 + \sigma^2\right) \\
= & F_i^{j,tE+0} + \left(\frac{L\eta^2}{2} - \eta\right)\left\|\nabla F_i^{j,tE+0}\right\|^2 + \frac{L\eta^2\sigma^2}{2}
\end{aligned}
\tag{21}
$$

The first inequality is based on Assumption 2, $\mathbb{E}[\langle \nabla F_i^{j,tE+0}, g_i^{j,tE+0}\rangle] = \|\nabla F_i^{j,tE+0}\|^2$. The second inequality is based on $Var(x) = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$. The third inequality is based on Assumption 3.

Taking Eq. (21) into Eq. (12), we further have:

$$
\begin{aligned}
\mathbb{E}\left[\Phi_i^{tE+1}\right] \le & \Phi_i^{tE+0} + \frac{L\eta^2\sigma^2}{2} + \left(\frac{L\eta^2}{2} - \eta\right)\lambda\left\|\nabla F_i^{tE+0}\right\|^2 \\
& + \left(\frac{L\eta^2}{2} - \eta\right)(1-\lambda)\left\|\nabla \mathcal{A}_i^{tE+0}\right\|^2
\end{aligned}
\tag{22}
$$

Taking expectation on both sides for E local epochs, we have:

$$
\begin{aligned}
\mathbb{E}\left[\Phi_i^{tE+E}\right] \le & \Phi_i^{tE+0} + \frac{LE\eta^2\sigma^2}{2} \\
& + \left(\frac{L\eta^2}{2} - \eta\right)\lambda\sum_{e=0}^{E-1}\left\|\nabla F_i^{tE+e}\right\|^2 \\
& + \left(\frac{L\eta^2}{2} - \eta\right)(1-\lambda)\sum_{e=0}^{E-1}\left\|\nabla \mathcal{A}_i^{tE+e}\right\|^2
\end{aligned}
\tag{23}
$$

□

Proof of Theorem 1. Based on Assumption 1, Assumption 2, Assumption 3, Assumption 4 and Lemma 1, we prove Theorem 1 as follows:

Proof. By telescoping Eq. (23) with Lemma 2 in [52], we have:

$$
\begin{aligned}
\mathbb{E}\left[\Phi_i^{(t+1)E+0}\right] \le & \Phi_i^{tE+0} + \frac{LE\eta^2\sigma^2}{2} + \eta\kappa^2 \\
& + \left(\frac{L\eta^2}{2} - \eta\right)\lambda\sum_{e=0}^{E-1}\left\|\nabla F_i^{tE+e}\right\|^2 \\
& + \left(\frac{L\eta^2}{2} - \eta\right)(1-\lambda)\sum_{e=0}^{E-1}\left\|\nabla \mathcal{A}_i^{tE+e}\right\|^2
\end{aligned}
\tag{24}
$$

We reformulate Eq. (24) into:

$$
\begin{aligned}
& \lambda\sum_{e=0}^{E-1}\left\|\nabla F_i^{tE+e}\right\|^2 + (1-\lambda)\sum_{e=0}^{E-1}\left\|\nabla \mathcal{A}_i^{tE+e}\right\|^2 \\
\le & \frac{\Phi_i^{tE+0} - \mathbb{E}\left[\Phi_i^{(t+1)E+0}\right] + \frac{LE\eta^2\sigma^2}{2} + \eta\kappa^2}{\eta - \frac{L\eta^2}{2}}
\end{aligned}
\tag{25}
$$

Let $\Delta = \Phi_i^{(T-1)E+0} - \Phi_i^{0E+0}$, we telescope Eq. (25) over T communication rounds, and have:

$$
\begin{aligned}
& \frac{1}{T}\sum_{t=0}^{T-1}\sum_{e=0}^{E-1}\left(\lambda\left\|\nabla F_i^{tE+e}\right\|^2 + (1-\lambda)\left\|\nabla \mathcal{A}_i^{tE+e}\right\|^2\right) \\
\le & \frac{\frac{\Delta}{T} + \frac{LE\eta^2\sigma^2}{2} + \eta\kappa^2}{\eta - \frac{L\eta^2}{2}}
\end{aligned}
\tag{26}
$$

We prove that for an arbitrary constant $\epsilon > 0$, as $T > 0$, $\Delta > 0$, Eq. (26) converges to ϵ when:

$$
\eta < \min\left(\frac{2(\epsilon - \kappa^2)}{L(\epsilon + E\sigma^2)}, \frac{2}{L}\right)
\tag{27}
$$

Consequently, when the learning rate η satisfies the condition in Eq. (27), FedAIMS converges. The convergence rate of FedAIMS is $O(1/T)$. □

References

1. Tan A Z, Yu H, Cui L, Yang Q. Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(12): 9587–9603
2. McMahan B, Moore E, Ramage D, Hampson S, Arcas y B A. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of International Conference on Artificial Intelligence and Statistics. 2017, 1273–1282

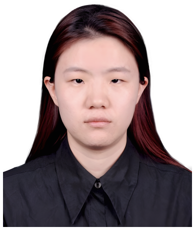
3. Leroy D, Coucke A, Lavril T, Gisselbrecht T, Dureau J. Federated learning for keyword spotting. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing. 2019, 6341–6345
4. Ouyang X, Xie Z, Zhou J, Huang J, Xing G. Clusterfl: a similarity-aware federated learning system for human activity recognition. In: Proceedings of Annual International Conference on Mobile Systems, Applications and Services. 2021, 54–66
5. Ouyang X, Shuai X, Li Y, Pan L, Zhang X, Fu H, Cheng S, Wang X, Cao S, Xin J, others . Admarker: A multimodal federated learning system for monitoring digital biomarkers of alzheimer's disease. In: Proceedings of Annual International Conference on Mobile Computing and Networking. 2024, 404–419
6. Sun N, Wang W, Tong Y, Liu K. Blockchain based federated learning for intrusion detection for internet of things. Frontiers of Computer Science, 2024, 18(5): 185328
7. Arivazhagan M G, Aggarwal V, Singh A K, Choudhary S. Federated learning with personalization layers. arXiv preprint arXiv:1912.00818, 2019
8. Liang P P, Liu T, Ziyin L, Allen N B, Auerbach R P, Brent D, Salakhutdinov R, Morency L P. Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523, 2020
9. Collins L, Hassani H, Mokhtari A, Shakkottai S. Exploiting shared representations for personalized federated learning. In: Proceedings of International Conference on Machine Learning. 2021, 2089–2099
10. Chen H Y, Chao W L. On bridging generic and personalized federated learning for image classification. In: Proceedings of International Conference on Learning Representations. 2022
11. Oh J, Kim S, Yun S Y. Fedbabu: Toward enhanced representation for federated image classification. In: Proceedings of International Conference on Learning Representations. 2022
12. Mclaughlin C, Su L. Personalized federated learning via feature distribution adaptation. In: Proceedings of Conference on Neural Information Processing Systems. 2024
13. Luo M, Chen F, Hu D, Zhang Y, Liang J, Feng J. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. Proceedings of Conference on Neural Information Processing Systems, 2021, 34: 5972–5984
14. Li Z, Shang X, He R, Lin T, Wu C. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In: Proceedings of International Conference on Computer Vision. 2023, 5319–5329
15. Lee C Y, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-supervised nets. In: Proceedings of International Conference on Artificial Intelligence and Statistics. 2015, 562–570
16. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2015, 1–9
17. Li T, Hu S, Beirami A, Smith V. Ditto: Fair and robust federated learning through personalization. In: Proceedings of International Conference on Machine Learning. 2021, 6357–6368
18. Zhang J, Hua Y, Wang H, Song T, Xue Z, Ma R, Guan H. Fedala: Adaptive local aggregation for personalized federated learning. In: Proceedings of AAAI Conference on Artificial Interlligence. 2023, 11237–11244
19. Zhang C J, Tong Y, Chen L. Where to: Crowd-aided path selection. Proceedings of the VLDB Endowment, 2014, 7(14): 2005–2016
20. Cao C C, Tong Y, Chen L, Jagadish H. Wisemarket: a new paradigm for managing wisdom of online social users. In: Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining. 2013, 455–463
21. Li T, Sahu A K, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. In: Proceedings of Conference on Machine Learning and Systems. 2020, 429–450
22. Li Q, He B, Song D. Model-contrastive federated learning. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 10713–10722
23. Wang Y, Tong Y, Shi D, Xu K. An efficient approach for cross-silo federated learning to rank. In: Proceedings of International Conference on Data Engineering. 2021, 1128–1139
24. Wang Y, Tong Y, Zhou Z, Ren Z, Xu Y, Wu G, Lv W. Fed-ltd: Towards cross-platform ride hailing via federated learning to dispatch. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 4079–4089
25. Tong Y, Zeng Y, Zhou Z, Liu B, Shi Y, Li S, Xu K, Lv W. Federated computing: Query, learning, and beyond. IEEE Data Eng. Bull., 2023, 46(1): 9–26
26. Wei S, Tong Y, Zhou Z, Xu Y, Gao J, Wei T, He T, Lv W. Federated reasoning llms: a survey. Frontiers of Computer Science, 2025, 19(12): 1912613
27. T Dinh C, Tran N, Nguyen J. Personalized federated learning with moreau envelopes. Proceedings of

Conference on Neural Information Processing Systems, 2020, 33: 21394–21405

28. Guo W, Zhuang F, Zhang X, Tong Y, Dong J. A comprehensive survey of federated transfer learning: challenges, methods and applications. Frontiers of Computer Science, 2024, 18(6): 186356
29. Zhang J, Guo S, Ma X, Wang H, Xu W, Wu F. Parameterized knowledge transfer for personalized federated learning. Proceedings of Conference on Neural Information Processing Systems, 2021, 34: 10092–10104
30. Sattler F, Müller K R, Samek W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(8): 3710–3722
31. Liu B, Ma Y, Zhou Z, Shi Y, Li S, Tong Y. Casa: Clustered federated learning with asynchronous clients. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024, 1851–1862
32. Huang Y, Chu L, Zhou Z, Wang L, Liu J, Pei J, Zhang Y. Personalized cross-silo federated learning on non-iid data. In: Proceedings of AAAI Conference on Artificial Interlligence. 2021, 7865–7873
33. Ye R, Ni Z, Wu F, Chen S, Wang Y. Personalized federated learning with inferred collaboration graphs. In: Proceedings of International Conference on Machine Learning. 2023, 39801–39817
34. Zhang W, Zhou Z, Wang Y, Tong Y. Dm-pfl: Hitchhiking generic federated learning for efficient shift-robust personalization. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 3396–3408
35. Chen D, Yao L, Gao D, Ding B, Li Y. Efficient personalized federated learning via sparse model-adaptation. In: Proceedings of International Conference on Machine Learning. 2023, 5234–5256
36. Sun Y, Wang X, Tang X. Deeply learned face representations are sparse, selective, and robust. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2015, 2892–2900
37. Wu X, Finnegan D, O'Neill E, Yang Y L. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In: Proceedings of The European Conference on Computer Vision. 2018, 237–253
38. Wang C, Wu Y, Chen S, Liu S, Li J, Qian Y, Yang Z. Improving self-supervised learning for speech recognition with intermediate layer supervision. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing. 2022, 7092–7096
39. Gou J, Yu B, Maybank S J, Tao D. Knowledge distillation: A survey. International Journal of Computer Vision, 2021, 129(6): 1789–1819
40. Romero A, Ballas N, Kahou S E, Chassang A, Gatta C, Bengio Y. Fitnets: Hints for thin deep nets. In: Proceedings of International Conference on Learning Representations. 2015
41. Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: Proceedings of International Conference on Learning Representations. 2017
42. Passalis N, Tefas A. Learning deep representations with probabilistic knowledge transfer. In: Proceedings of The European Conference on Computer Vision. 2018, 268–284
43. Kim J, Park S, Kwak N. Paraphrasing complex network: Network compression via factor transfer. Proceedings of Conference on Neural Information Processing Systems, 2018, 31
44. Tun Y L, Thwal C M, Park Y M, Park S B, Hong C S. Federated learning with intermediate representation regularization. In: Proceedings of International Conference on Big Data and Smart Computing. 2023, 56–63
45. Li R, Wang X, Huang G, Yang W, Zhang K, Gu X, Tran S N, Garg S, Alty J, Bai Q. A comprehensive review on deep supervision: Theories and applications. arXiv preprint arXiv:2207.02376, 2022
46. Tan Y, Long G, Liu L, Zhou T, Lu Q, Jiang J, Zhang C. Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of AAAI Conference on Artificial Interlligence. 2022, 8432–8440
47. Xu J, Tong X, Huang S L. Personalized federated learning with feature alignment and classifier collaboration. In: Proceedings of International Conference on Learning Representations. 2023
48. Yang J, Duan Y, Qiao T, Zhou H, Wang J, Zhao W. Prototyping federated learning on edge computing systems. Frontiers of Computer Science, 2020, 14(6): 146318
49. Qu L, Li S, Zhou Z, Liu B, Xu Y, Tong Y. Darkdistill: Difficulty-aligned federated early-exit network training on heterogeneous devices. In: Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2025
50. Li X, Huang K, Yang W, Wang S, Zhang Z. On the convergence of fedavg on non-iid data. In: Proceedings of International Conference on Learning Representations. 2020
51. Zeiler M D, Fergus R. Visualizing and understanding convolutional networks. In: ECCV. 2014, 818–833
52. Yi L, Yu H, Ren C, Wang G, Li X, others . Federated

model heterogeneous matryoshka representation learning. In: Proceedings of Conference on Neural Information Processing Systems. 2024

53. Cohen G, Afshar S, Tapson J, Van Schaik A. Emnist: Extending mnist to handwritten letters. In: Proceedings of International Joint Conference on Neural Networks. 2017, 2921–2926
54. Krizhevsky A, Hinton G, others . Learning multiple layers of features from tiny images, 2009
55. Chrabaszcz P, Loshchilov I, Hutter F. A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819, 2017
56. Hsu T M H, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335, 2019
57. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2016, 770–778



Shuyuan Li recieved her Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China in 2024. She is currently a postdoc at the Department of Data Science, City University of Hong Kong. Her research interests include federated computing, privacy preserving data analytics.

Boyi Liu received his B.E. degree from the School of Computer Science and Engineering, Beihang University, Beijing, China in 2023. He is currently a Ph.D. student at the School of Computer Science and Engineering, Beihang University, China. He is also a joint Ph.D. student at the Department of Data Science, City University of Hong Kong, China. His research interests include federated learning.

Zimu Zhou received the B.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011, and the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2015. He is currently an assistant professor at the Department of Data Science, City University of Hong Kong. His research focuses on ubiquitous computing.

Jin Dong received the Ph.D. degree in from the Department of Automation, Tsinghua University, Beijing, China, in 2011. He is the General Director of Beijing Academy of Blockchain and Edge Computing. He is also the General Director of Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing. The team he led developed “ChainMaker”, the first hardware-software integrated blockchain system around the globe. He has been dedicated in the research areas such as blockchain, AI and low-power chip design.