A Fair Federated Learning Method for Handling Client Participation Probability Inconsistencies in Heterogeneous Environments

Siyuan Wu[1], **Yongzhe Jia**[1], **Haolong Xiang**[2], **Xiaolong Xu**[2],
Xuyun Zhang[3], **Lianyong Qi**[4], **Wanchun Dou**[1,*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China
[2]School of Computer and Software,
Nanjing University of Information Science and Technology, China
[3]School of Computing, Macquarie University, Australia
[4]College of Computer Science and Technology, China University of Petroleum (East China), China

Abstract

Federated learning (FL) is a distributed machine learning paradigm that enables multiple clients to collaboratively train a shared model without exposing their raw data. However, existing FL research has primarily focused on optimizing learning performance based on the assumption of uniform client participation, with few studies delving into performance fairness under inconsistent client participation, particularly in model-heterogeneous FL environments. In view of this challenge, we propose **PHP-FL**, a novel model-heterogeneous FL method that explicitly addresses scenarios with varying client participation probabilities to enhance both model accuracy and performance fairness. Specifically, we introduce a Dual-End Aligned ensemble Learning (DEAL) module, where small auxiliary models on clients are used for dual-end knowledge alignment and local ensemble learning, effectively tackling model heterogeneity without a public dataset. Furthermore, to mitigate update conflicts caused by inconsistent participation probabilities, we propose an Importance-driven Selective Parameter Update (ISPU) module, which accurately updates critical local parameters based on training progress. Finally, we implement PHP-FL on a lightweight FL platform with heterogeneous clients across three different client participation patterns. Extensive experiments under heterogeneous settings and diverse client participation patterns demonstrate that PHP-FL achieves state-of-the-art performance in both accuracy and fairness. Our code is available at: `https://github.com/Siyuan01/PHP-FL-main`.

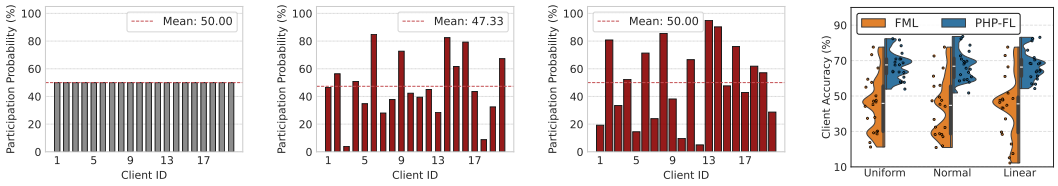
1 Introduction

Federated Learning (FL) has emerged as a promising paradigm for enabling decentralized model training across multiple clients without directly sharing their private data [1, 2]. By collaboratively learning a global model while keeping data localized, FL offers strong privacy guarantees and broad applicability across various domains such as mobile devices, healthcare, and finance [3–5].

Despite significant progress, traditional FL methods still face two critical challenges: **I) Model heterogeneity.** Traditional FL assumes that all clients share an identical local model architecture, which is often impractical in real-world due to the diversity in client capabilities. To address this, Model-Heterogeneous Federated Learning (MH-FL) has emerged as a promising research paradigm [2, 6–9], which allows each client to maintain a personalized model tailored to its own resource constraints

*Corresponding author: Wanchun Dou (douwc@nju.edu.cn)

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) ***Uniform*** pattern (b) ***Normal*** pattern (c) ***Linear*** pattern (d) Client accuracy

Figure 1: Left three: three client participation probability patterns (average value is approximately 50%). Right: client accuracy under patterns (a)-(c). Refer to Section 5.1 for experimental details.

or task requirements. However, most existing MH-FL methods [10–13] primarily aim to ensure compatibility between diverse model architectures, yet overlook the fairness issues posed arising from inconsistent client participation probabilities in practical deployments. This oversight can result in biased models that favor frequently participating clients, ultimately compromising the robustness and fairness of the FL system. **II) Unfairness caused by inconsistent client participation.** Most existing FL research [1, 9, 12–16] implicitly assume a uniform client participation pattern, where all clients are equally likely to participate in each training round, as illustrated in Figure 1a. In practical deployments, clients often face heterogeneous conditions, such as intermittent connectivity, fluctuating network bandwidth, and network coverage of base stations [17–20]. These internal and external factors lead to client unavailability and result in non-uniform participation probabilities, as exemplified in Figure 1b and 1c, which may critically degrade both the overall accuracy and fairness of the FL system. Figure 1d compares the final client accuracy distributions between FML [21] (a representative MH-FL method) and our method PHP-FL under three participation patterns. While FML suffers significant performance degradation in the *normal* and *linear* patterns compared to the *uniform* participation, PHP-FL maintains stable accuracy with only marginal drops. Furthermore, PHP-FL demonstrates tighter accuracy distributions, indicating better performance fairness. Although some studies [22–25] improve overall system performance by proactively selecting high-availability clients and discarding less efficient ones, such strategies often compromise fairness across clients. Furthermore, numerous fair federated learning methods aim to enhance performance fairness through personalized model [14, 26] or weight recalibration [27, 28]. Nevertheless, they typically operate under the assumption of uniform client participation, failing to address the challenge of inconsistent client availability. Despite its importance, this issue has received limited attention in the literature, particularly in the context of MH-FL, where the interplay between model heterogeneity and participation imbalance exacerbates the learning challenge.

To address these challenges, we propose PHP-FL, a fair federated learning method designed to address scenarios with varying client participation probabilities in model-heterogeneous environments. Specifically, to effectively tackle model heterogeneity without relying on public datasets, we introduce a Dual-End Aligned ensemble Learning (DEAL) module, which leverages lightweight auxiliary models on clients to align heterogeneous local models and enable ensemble learning to improve the performance of local tasks. Furthermore, to mitigate the adverse effects of update conflicts that caused by inconsistent client participation probabilities, we propose a Importance-driven Selective Parameter Update (ISPU) module. The ISPU module adaptively updates only the most critical task-relevant parameters based on training progress, allowing clients with different participation frequencies to selectively absorb varying ratios of global knowledge. This design helps reduce gradient conflicts and enhancing fairness. Our main contributions are as follows:

- To the best of our knowledge, this is the first work to explicitly address performance unfairness caused by inconsistent client participation probabilities in practical FL systems with heterogeneous local models.
- We propose PHP-FL, a novel model-heterogeneous federated learning method designed to address inconsistent client participation probabilities, aiming to jointly improve both overall accuracy and performance fairness across clients.
- We evaluate PHP-FL through extensive experiments on a lightweight FL platform simulating multiple realistic participation patterns. Empirical results on the Fashion-MNIST and CIFAR-10 datasets demonstrate its state-of-the-art performance, while the ablation study further validates the effectiveness of each proposed module.

2 Related Works

Model-Heterogeneous Federated Learning. Model-Heterogeneous Federated Learning (MH-FL) has emerged as a promising research direction [11, 6, 8, 9, 2, 13]. Existing work in this area can be broadly categorized into *knowledge distillation*-based (KD) methods, *representation alignment* methods, and *partial model sharing* methods. KD is one of the most widely adopted techniques in MH-FL. The studies in [29–31] enable clients with different architectures to distill knowledge through a shared or public dataset. However, such reliance on public data limits applicability in privacy-sensitive settings. Another popular line of work, like [15, 12, 16], focuses on representation alignment, which aligns feature representations or prototypes rather than raw model parameters, allowing clients to maintain model diversity while contributing to a shared learning objective. Furthermore, some studies [7, 32–34] adopt partial model sharing strategies, where clients share only specific model components (*e.g.*, a shared backbone or predictor) while keeping the other parts distinct, thereby enabling partial compatibility across models. However, few MH-FL methods explicitly address fairness for clients under inconsistent participation probabilities, a critical requirement for equitable and robust deployment.

Fairness in Federated Learning. Existing research on fairness in federated learning has primarily focused on three key dimensions: (1) *Contribution Fairness* [4, 35, 36], which involves evaluating each client's contribution to the global model to guide equitable benefit distribution, often using techniques like Shapley value or influence functions [37, 38]; (2) *Model Fairness* [39, 40], which addresses inherent biases in model predictions concerning sensitive attributes, thereby promoting fairness at the prediction level; and (3) *Performance Fairness* [14, 20, 27, 41, 28], which aims to ensure uniform model performance across clients, typically by minimizing the variance or standard deviation of test accuracy among clients. As prior studies [42, 43] demonstrate, these fairness metrics often conflict, making it infeasible for a model to simultaneously achieve optimal performance across all dimensions. Our work, therefore, specifically targets performance fairness, aiming to ensure uniform performance across clients while concurrently optimizing overall performance under inconsistent participation probabilities in MH-FL. While another line of research [20, 44, 45] addresses client unavailability by primarily focusing on maintaining the average performance across clients, they do not explicitly ensure performance-level fairness. Furthermore, these methods are typically designed for homogeneous settings and face significant challenges when generalizing to heterogeneous federated learning environments, where client capabilities vary substantially.

3 Preliminaries

The Global Objective of Federated Learning. Following typical federated learning [1, 26] settings, we consider a set of K clients (index by i) with local datasets $\{D_1, D_2, ..., D_K\}$, where $D_i = \{(x_j, y_j)\}_{j=1}^{n_i}$ and $n_i = |D_i|$. In a heterogeneous FL environment, each client i maintains a unique model $\boldsymbol{w}_i$, parameterized by $\theta_i \in \mathbb{R}^{d_i}$, with dimensional heterogeneity ($d_i \neq d_j$) arising from hardware constraints (*e.g.*, compute/memory limits) or personalized model specialization. This implies $\dim(\theta_i) \neq \dim(\theta_j)$, $\exists i \neq j \in [K]$. The global objective function can be expressed as:

$$\min_{\{\boldsymbol{w}_i\}_{i=1}^K} F(\{\boldsymbol{w}_i\})_{i=1}^K = \sum_{i=1}^K p_i F_i(\boldsymbol{w}_i), \quad \sum_{i=1}^K p_i = 1, \tag{1}$$

where p_i is the weight of client i, $\{\boldsymbol{w}_i\}_{i=1}^K$ represents the set of the client's local models, and $F_i(\boldsymbol{w}_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{L}(\boldsymbol{w}_i(\theta_i; x_j), y_j)$ is the local objective for client i with loss function $\mathcal{L}$.

Inconsistent Client Participation Probability. To simulate realistic client availability in federated learning, we consider three types of client participation probability patterns. Let $p_{i,t}$ denote the probability that client i actively participates in communication round t:

Definition 1 *(Uniform Pattern) All clients share an identical and fixed probability $a \in (0, 1]$ of participating in each round , i.e., $p_{i,t} = a, \quad \forall i \in \{1, 2, \ldots, n\}, \ \forall t$.*

Definition 2 *(Normal Pattern) The participation probabilities are drawn from a truncated normal distribution to simulate natural heterogeneity: $p_{i,t} \sim \mathcal{N}(\mu, \sigma)$, and $p_{i,t}$ is clipped to $(0, 1]$.*

Definition 3 *(Linear Pattern) Client participation probabilities are distributed according to an increasing arithmetic sequence:* $p_{i,t} = a + (i-1)d, \quad i = 1, 2, \ldots, n, \ \forall t$ *, where* a *is the first term and* d *is the common difference. In this pattern, the initial sequence satisfies* $0 < p_{1,t} < p_{2,t} < \cdots < p_{K,t} \leq 1$ *for round* t*. This sequence* $\{p_{i,t}\}_{i=1}^{K}$ *is then randomly shuffled prior to use to eliminate any inherent ordering bias among clients. It models systematic heterogeneity such as time-varying connectivity or device capacity.*

Note that $p_{i,t}$ is independent of the history and other clients. The three distinct patterns considered enable a comprehensive analysis of how heterogeneous client participation affects both fairness and overall performance in federated learning.

Design Goals. In this paper, we aim to design a MH-FL method under inconsistent client participation probabilities that not only optimizes the *average performance* across all clients but also enhances *performance fairness*. Formally, let $a_i (i = 1, \ldots, K)$ represent the test accuracy on the i-th client's local test dataset. The Accuracy Metric (AM) is defined as: $AM = \frac{1}{K} \sum_{k=1}^{K} a_k$. The Fairness Metric (FM) is defined as: $FM = \mathrm{Std}(a_1, \ldots, a_K)$, where $\mathrm{Std}(\cdot)$ denotes the standard deviation. To this end, our approach seeks to maximize the average local accuracy (AM) while minimizing the performance disparity (FM), ensuring both high overall performance and fair performance distribution across clients.

4 Methodology

4.1 Overview of PHP-FL

Our method consists of two key modules: dual-end aligned ensemble learning (DEAL in Section 4.2), and importance-driven selective parameter update (ISPU in Section 4.3). The DEAL module employs a small homogeneous auxiliary model to perform bidirectional representation-logit alignment between local and auxiliary models, which resolves model heterogeneity and enhances overall performance via ensemble learning. To handle inconsistent client participation, the ISPU module selectively updates task-relevant critical parameters using an importance-based masking mechanism. This approach applies larger updates to stragglers to accelerate overall convergence, while reducing updates for frequent participants to prevent adverse effects from the stragglers. This ensures efficient knowledge fusion and fair parameter evolution across clients.

As shown in Figure 2, the training process in each communication round t of PHP-FL can be summarized as follows:[2] ❶ The server first computes the client-specific auxiliary model $\mathcal{G}^{t-1} \odot M_i^h$ for the active client $i \in \mathcal{A}_t$ by pruning non-essential parameters using historical binary mask M_i^h. ❷ Then active clients initialize the personalized auxiliary models $\hat{\boldsymbol{g}}_i^{t-1}$. ❸ PHP-FL decomposes both the local model $\boldsymbol{w}_i$ and the auxiliary model $\boldsymbol{g}_i$ into a *backbone* and a *predictor*, which are used for representation extraction and soft prediction, respectively. At the beginning of local training, the DEAL module first optimizes the ensemble weights $\boldsymbol{\lambda}_i$ on the adaptation set $\mathcal{D}_i^a$, while $\boldsymbol{w}_i^t$ and $\hat{\boldsymbol{g}}_i^{t-1}$ are frozen. ❹ Local training then proceeds using the customized loss $\mathcal{L}_{\boldsymbol{w}}$, which comprises two components: (1) The *dual-end alignment loss* $\mathcal{L}_{\text{DEAL}}$, enabling bidirectional knowledge alignment through data-free distillation and representation matching. (2) The *ensemble learning loss* $\mathcal{L}_{\text{ENS}}$, which further enhances overall model performance. ❺ Following the local training, the ISPU module calculates an update ratio α_i^t based on the client's total training rounds. It then estimates the top-α_i^t important parameters of $\boldsymbol{g}_i^t$ using the ℓ_1-norm and generates a binary mask M_i^t. Finally, each active client uploads both its updated local auxiliary model $\boldsymbol{g}_i^t$ and the binary mask M_i^t to the server. ❻ The server updates the historical mask $M_i^h \in M_{\text{hist}}$ for each active client i by replacing its entry with the newly received M_i^t. Then the received auxiliary models are aggregated via a simple averaging technique to obtain the global auxiliary model $\mathcal{G}^t$ for round $t+1$.

4.2 Dual-End Aligned Ensemble Learning

To address system heterogeneity without relying on public datasets, previous works such as [12, 15] decompose the model w[3] (parameterized by θ) into a *backbone* $\boldsymbol{w}_b$ and a *predictor* $\boldsymbol{w}_p$, and perform

[2] Algorithm 1 in Appendix A describes the PHP-FL algorithm.

[3] We omit the client index i and the communication round index t for notation simplicity.

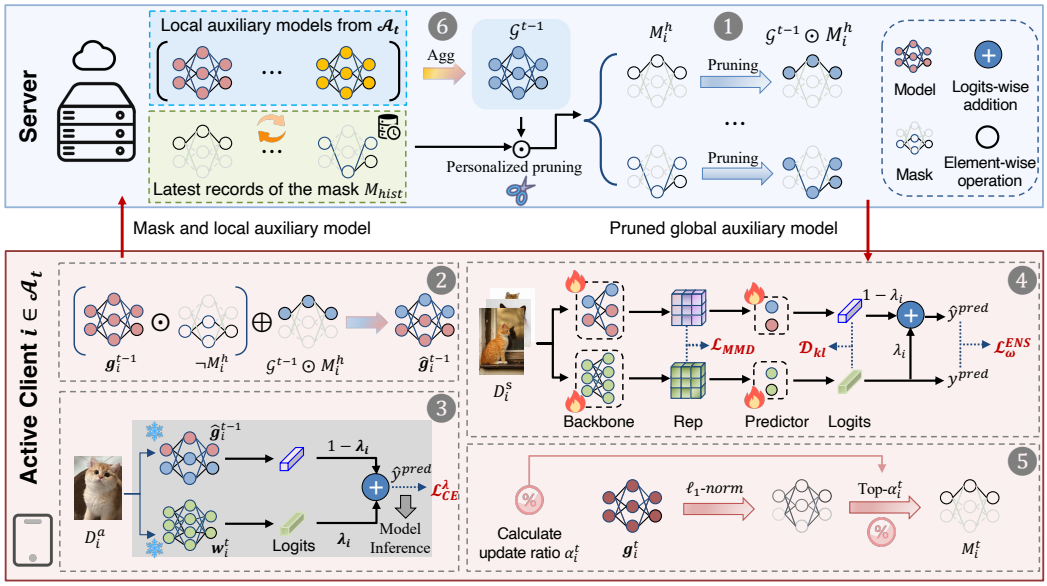


Figure 2: The overview of PHP-FL.

aggregation based on intermediate representations $z = \boldsymbol{w}_b(\theta_b; x_j)$ produced by the backbone. However, it's challenging to classify representations generated by heterogeneous backbones for local predictors. Inspired by the spirit of the mutual learning, some works [46, 47] leverage logits-level knowledge distillation, where each client co-trains a heterogeneous local model $\boldsymbol{w}$ and a lightweight homogeneous global model $\boldsymbol{g}$ (parameterized by ϕ) by aligning only the final outputs of the predictors. Unfortunately, this limited alignment fails to facilitate meaningful knowledge transfer to the backbone component, resulting in suboptimal representation learning. To address these limitations, we propose a DEAL module. In DEAL, both the backbone and predictor components between local and auxiliary models are explicitly aligned. This dual-end alignment ensures effective and rapid fusion of local and global knowledge. To this end, we design the following loss function:

$$\mathcal{L}_{DEAL}^{\boldsymbol{w}} = \frac{1}{|D_i|} \sum_{j \in D_i} [\underbrace{\mathcal{L}_{\text{MMD}}(\boldsymbol{w}_b(\theta_b; x_j), \boldsymbol{g}_b(\phi_b; x_j))}_{\text{Backbone Alignment}} + \underbrace{\mathcal{D}_{\text{KL}}(\boldsymbol{w}(\theta; x_j) \| \boldsymbol{g}(\phi; x_j))}_{\text{Predictor Alignment}}], \tag{2}$$

where $\mathcal{D}_{\text{KL}}$ is Kullback-Leibler (KL) divergence. The Maximum Mean Discrepancy (MMD) loss $\mathcal{L}_{\text{MMD}}$ between two sets of representations $\mathbf{z}_1 \in \mathbb{R}^{n \times d_1}$ and $\mathbf{z}_2 \in \mathbb{R}^{m \times d_2}$ using a Gaussian radial basis function (RBF) kernel is computed as:

$$\begin{aligned} \mathcal{L}_{\text{MMD}}(\mathbf{z}_1, \mathbf{z}_2) = & \frac{1}{n^2} \sum_{i,j=1}^{n} k(f(\mathbf{z}_1^{(i)}), f(\mathbf{z}_1^{(j)})) + \frac{1}{m^2} \sum_{i,j=1}^{m} k(h(\mathbf{z}_2^{(i)}), h(\mathbf{z}_2^{(j)})) \\ & - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(f(\mathbf{z}_1^{(i)}), h(\mathbf{z}_2^{(j)})), \end{aligned} \tag{3}$$

where $f : \mathbb{R}^{d_1} \to \mathbb{R}^d$ and $h : \mathbb{R}^{d_2} \to \mathbb{R}^d$ represent customizable projection functions designed to standardize feature dimensions, enabling cross-architecture feature alignment between the local model $\boldsymbol{w}$ and global model $\boldsymbol{g}$ when their structures are heterogeneous. The Gaussian RBF kernel $k(\mathbf{z}, \mathbf{z}') = \exp\left(-\gamma \|\mathbf{z} - \mathbf{z}'\|^2\right)$ measures similarity between representations, with $\gamma = 1/(2\sigma^2)$ controlling the kernel bandwidth. This non-parametric metric effectively captures the distance between the distributions of $\mathbf{z}_1$ and $\mathbf{z}_2$ in the Reproducing Kernel Hilbert space (RKHS). MMD is able to effectively align feature distributions by comparing global statistics via kernel-based embeddings, handles non-IID data robustly, and enables stable optimization [48].

Next, to fully leverage the classification capabilities of both local models and global auxiliary models, we adopt model ensembling [49, 47, 13] to enhance performance on local tasks:

$$\mathcal{L}_{ENS}^{\boldsymbol{w}} = \frac{1}{|D_i|} \sum_{j \in D_i} [\mathcal{L}_{\text{CE}}(\boldsymbol{w}(\theta; x_j), y_j) + \mathcal{L}_{\text{CE}}(\lambda \boldsymbol{w}(\theta; x_j) + (1-\lambda)\boldsymbol{g}(\phi; x_j), y_j)], \tag{4}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss between the predicted label and the ground-truth label. Furthermore, to address the challenges posed by the potential heterogeneous system capabilities between each local model $\boldsymbol{w}$ and the global model $\boldsymbol{g}$, we set $\boldsymbol{\lambda}$ as a trainable parameter for each client and randomly hold out a tiny *adaptability set* D_i^a from the training set D_i (*e.g.*, 10%) for its optimization at the beginning of every communication round. The remaining set on each client for training is denoted as the *study set* D_i^s. This round-wise resampling of the adaptability set guarantees that the ensemble weight $\boldsymbol{\lambda}_i$ is continuously optimized on fresh and unbiased data, thereby effectively mitigating overfitting risks. The learning process of $\boldsymbol{\lambda}$ on each client is:

$$\boldsymbol{\lambda}^t \leftarrow \boldsymbol{\lambda}^{t-1} - \eta_{\boldsymbol{\lambda}} \frac{1}{|D_i^a|} \sum_{j \in D_i^a} \nabla_{\boldsymbol{\lambda}^{t-1}} \mathbb{E}_{(x_j, y_j) \sim D_i^a} \mathcal{L}_{\text{CE}}(\boldsymbol{\lambda}^{t-1} \boldsymbol{w}(\theta; x_j) + (1 - \boldsymbol{\lambda}^{t-1}) \boldsymbol{g}(\phi; x_j), y_j), \tag{5}$$

where $\eta_{\boldsymbol{\lambda}}$ is the learning rate for $\boldsymbol{\lambda}$. Finally, the total training objective of the local model $\boldsymbol{w}$ combines the dual-end alignment loss and the ensemble learning loss:

$$\mathcal{L}_{\boldsymbol{w}} = \mathcal{L}_{DEAL}^{\boldsymbol{w}} + \mathcal{L}_{ENS}^{\boldsymbol{w}}, \tag{6}$$

For symmetry, an analogous loss $\mathcal{L}_{\boldsymbol{g}}$ is also computed but omitted here for brevity, as it follows the same formulation with reversed inputs. The total losses $\mathcal{L}_{\boldsymbol{g}}$ and $\mathcal{L}_{\boldsymbol{w}}$ are used to simultaneously update the homogeneous auxiliary model and the heterogeneous client local model, with learning rates $\eta_{\boldsymbol{g}}$ and $\eta_{\boldsymbol{w}}$, respectively, as follows:

$$\begin{aligned}
\boldsymbol{w}^t &\leftarrow \boldsymbol{w}^{t-1} - \eta_{\boldsymbol{w}} \frac{1}{|D_i^s|} \sum_{j \in D_i^s} \nabla_{\boldsymbol{w}^{t-1}} \mathbb{E}_{(x_j, y_j) \sim D_i^s} \mathcal{L}_{\boldsymbol{w}}(\boldsymbol{w}^{t-1}, \boldsymbol{g}^{t-1}, \boldsymbol{\lambda}^t, x_j, y_j), \\
\boldsymbol{g}^t &\leftarrow \boldsymbol{g}^{t-1} - \eta_{\boldsymbol{g}} \frac{1}{|D_i^s|} \sum_{j \in D_i^s} \nabla_{\boldsymbol{g}^{t-1}} \mathbb{E}_{(x_j, y_j) \sim D_i^s} \mathcal{L}_{\boldsymbol{g}}(\boldsymbol{g}^{t-1}, \boldsymbol{w}^{t-1}, \boldsymbol{\lambda}^t, x_j, y_j).
\end{aligned} \tag{7}$$

During the inference stage, clients use the weighted model ensemble for prediction:

$$\hat{y}_j^{pred} = \arg\max(\boldsymbol{\lambda} \boldsymbol{w}(\theta; x_j) + (1 - \boldsymbol{\lambda}) \boldsymbol{g}(\phi; x_j)). \tag{8}$$

This adaptive weighting mechanism automatically balances the contributions of both models based on their current performance. It is particularly effective under system heterogeneity, where devices may have varying computational capabilities.

4.3 Importance-Driven Selective Parameter Update

To enable straggling clients to quickly catch up upon rejoining training while preserving the learning momentum of more active clients, we propose a novel selective parameter update module, ISPU, which selectively updates task-relevant parameters and suppresses noisy or redundant updates. Instead of directly overwriting the local auxiliary model $\boldsymbol{g}_i^{t-1}$ with the global model $\mathcal{G}^{t-1}$, we perform a model fusion by identifying the most significant parameters in $\boldsymbol{g}_i^{t-1}$. The update ratio α_i^t is adaptively determined by a sigmoid-based scheduling function according to the training progress:

$$\alpha_i^t = \frac{1}{1 + \exp\left(\delta \cdot \left(\frac{N_i(t)}{t+1} - 0.5\right)\right)} \cdot \tau \tag{9}$$

where $N_i(t) = \sum_{r=1}^{t} \mathbb{1}\{i \in \mathcal{A}_r\}$ denotes the cumulative number of rounds in which client i has participated up to round t and $\mathbb{1}\{\cdot\}$ is the indicator function. Here, $\tau \in (0, 1]$ represents the pruning threshold and δ is a tunable sharpness hyperparameter. We then apply a binary mask on the parameters of $\boldsymbol{g}_i^{t-1}$ to retain only the critical parameters and replace them with the corresponding parameters from the global model $\mathcal{G}^{t-1}$. Common pruning metrics include the ℓ-norm [50, 3], Fisher Information Matrix (FIM) [51, 52], and sensitivity-based measures [53, 54]. Specifically, we adopt

the ℓ_1-norm to evaluate the importance of parameters, which has been proven to be an effective technique for assessing parameter significance [55, 3]. Compared to other metrics, this formulation better captures the importance of task-relevant parameters. After local training, the binary mask M_i^t is constructed to update only the top-α_i^t important parameters of $\boldsymbol{g}_i^t$ in the next participation round:

$$M_{i,d}^t = \begin{cases} 1, & \text{if } d\text{-th parameter } \in \text{top-}\alpha_i^t \text{ largest of } \boldsymbol{g}_i^t \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

This parameter-wise filtering mechanism helps **frequently active clients** preserve their learned knowledge while allowing **infrequent clients** to assimilate global updates more effectively, enabling rapid catch-up and mitigating knowledge drift. Specifically, at the beginning of round t, the local auxiliary model $\boldsymbol{g}_i^{t-1}$ of each active client $i \in \mathcal{A}_t$ is updated as follows:

$$\hat{\boldsymbol{g}}_i^{t-1} = \boldsymbol{g}_i^{t-1} \odot \neg M_i^h + \mathcal{G}^{t-1} \odot M_i^h, \tag{11}$$

where M_i^h is the mask matrix obtained by client i from its most recent training round and $\neg M$ denotes the bit-wise inverse of the mask M. This approach ensures clients preserve critical knowledge via high-importance parameters, filters out conflicts from heterogeneous data and inconsistent training progress.

5 Experiments

5.1 Experiments Details

Datasets. We evaluate our proposed PHP-FL on two standard image classification benchmarks: Fashion-MNIST[4] [56] and CIFAR-10[5] [57]. For both datasets, we adopt a 4:1 ratio to split samples into training and test sets. Following previous studies [12, 58, 59], we simulate heterogeneous data distributions by allocating class j proportions to each client k according to a Dirichlet distribution ($p_{j,k} \sim \text{Dir}(\beta)$), where a smaller β implies more extreme data heterogeneity across clients. We adopt $\beta = 0.1$ for Fashion-MNIST and $\beta = 0.5$ for CIFAR-10, respectively. Notably, each client's local training and test sets share the same distribution.

Models. Our experimental setup employs four heterogeneous model architectures: (1) GoogleNet [60], (2) DenseNet-121 [61], (3) EfficientNet-B1 [62], and (4) ResNet-18 [63]. Each client is assigned one of these models based on its identifier i, following a round-robin strategy where client i receives the model corresponding to $i \bmod 4$. Comprehensive comparative results of homogeneous model architectures are provided in Appendix C.2.

Comparison Baselines. In the heterogeneous model experiments, we comprehensively evaluate our method against six state-of-the-art heterogeneous federated learning algorithms without relying on public data, including FML [21], FedGen [10], FedKD [11], FedAPEN [47], FedTGP [12], FedMRL [13]. In addition, we also compare a standalone baseline where clients train locally without any aggregation or communication.

Implementation Details. Our experimental framework is built on a lightweight MH-FL platform HtFLlib [64] using PyTorch 2.2.2 [65] with NVIDIA RTX 3090 GPU. We employ SGD as our optimizer with a learning rate of 0.001 and a local batch size of 64. The global training process consists of 100 communication rounds, with a total of 20 clients. During each federated training round, clients perform 5 local epochs of training. In each round, client participation follows the three patterns introduced in Section 3. We repeated all experiments three times with different random seeds and present the averaged results. More details are provided in Appendix B.

Evaluation Metrics. As defined in Section 3, we evaluate the performance using the average Top-1 accuracy (AM) across all clients. In evaluating the fairness of the clients, we adopt the standard deviation (FM) of client accuracy when the Top-1 test accuracy is achieved. In PHP-FL, the test accuracy and fairness for each client are derived from the ensemble output of its local and auxiliary models, as computed by Eq. 8.

[4] `https://github.com/zalandoresearch/fashion-mnist?tab=readme-ov-file`

[5] `https://www.cs.toronto.edu/~kriz/cifar.html`

Table 1: Comparison with the state-of-the-art methods on Fashion-MNIST in the heterogeneous setting. Best in **bold** and second with underline. ↑ indicates improved accuracy (%) and ↓ indicates improved standard deviation (%) of accuracy compared with the best baseline, respectively.

Methods	***Uniform*** $[a = 0.5]$		***Normal*** $[\mu = 0.5, \sigma = 0.2]$		***Linear*** $\left[a = 0.05, d = \frac{K-2}{K(K-1)}\right]$	
	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓
Standalone	95.88±0.19	9.26±1.37	96.89±0.19	9.37±1.22	95.77±0.05	9.86±0.53
FML [arXiv20]	89.09±0.66	23.13±1.18	88.71±0.23	23.32±1.11	88.15±0.73	25.21±2.76
FedGen [ICML21]	93.97±1.13	22.78±1.46	93.81±0.99	23.25±1.52	93.72±0.93	23.80±1.90
FedKD [NC22]	95.67±0.31	9.20±0.92	95.65±0.30	9.02±1.16	95.57±0.18	9.30±0.78
FedAPEN [KDD23]	96.79±0.06	6.74±0.26	96.79±0.07	6.50±0.60	96.73±0.06	6.89±0.08
FedTGP [AAAI24]	94.35±2.57	10.94±2.47	94.06±2.38	11.32±2.25	94.21±2.48	11.50±2.18
FedMRL [NIPS24]	96.06±0.45	9.07±1.65	96.05±0.43	9.25±1.39	95.78±0.09	9.81±0.64
PHP-FL (Ours)	**97.64±0.04**	**4.15±0.18**	**97.59±0.04**	**4.24±0.05**	**97.58±0.03**	**4.29±0.04**
	↑ 0.85	**↓ 2.59**	**↑ 0.80**	**↓ 2.26**	**↑ 0.95**	**↓ 2.60**

Table 2: Comparison with the state-of-the-art methods on CIFAR-10 in the heterogeneous setting. Best in **bold** and second with underline. ↑ indicates improved accuracy (%) and ↓ indicates improved standard deviation (%) of accuracy compared with the best baseline, respectively.

Methods	***Uniform*** $[a = 0.5]$		***Normal*** $[\mu = 0.5, \sigma = 0.2]$		***Linear*** $\left[a = 0.05, d = \frac{K-2}{K(K-1)}\right]$	
	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓
Standalone	56.66±0.32	10.28±0.32	56.77±0.40	10.19±0.36	56.61±0.30	10.21±0.32
FML [arXiv20]	46.61±0.61	15.18±0.34	46.16±1.02	15.78±0.75	46.04±1.16	15.85±0.83
FedGen [ICML21]	54.27±0.15	10.29±0.26	54.38±0.02	10.33±0.31	54.50±0.18	10.45±0.46
FedKD [NC22]	54.86±0.43	10.43±0.23	54.82±0.37	10.36±0.33	54.55±0.13	10.45±0.21
FedAPEN [KDD23]	60.30±0.33	9.40±0.30	60.35±0.31	9.45±0.31	60.40±0.31	9.40±0.30
FedTGP [AAAI24]	53.39±0.53	9.73±0.61	53.40±0.52	10.62±1.17	52.85±1.16	10.28±0.78
FedMRL [NIPS24]	52.80±0.63	12.81±1.17	52.61±0.77	12.85±1.20	53.11±0.63	12.95±1.27
PHP-FL (Ours)	**66.85±0.36**	**8.07±0.25**	**66.94±0.39**	**8.07±0.24**	**66.78±0.36**	**8.09±0.27**
	↑ 6.55	**↓ 1.38**	**↑ 6.59**	**↓ 1.29**	**↑ 6.38**	**↓ 1.31**

5.2 Comparison to State-of-the-Arts Methods

We evaluate PHP-FL against several state-of-the-art methods in Tables 1 and 2. The experiments are conducted under three distinct client participation patterns: *uniform*, *normal*, and *linear*, representing diverse real-world scenarios. Across all settings, PHP-FL consistently demonstrates superior performance. It achieves the highest average accuracy (highest AM values) while simultaneously exhibiting the best fairness (lowest FM values). Compared to the strongest baseline FedAPEN, PHP-FL achieves notable improvements in average performance, boosting AM by up to 0.87% on Fashion-MNIST and a substantial 6.51% on CIFAR-10. At the same time, it enhances fairness by reducing FM by up to 2.48% and 1.33% on the respective datasets, demonstrating the robustness and effectiveness of PHP-FL in addressing model heterogeneity and unfairness caused by inconsistent client participation. Appendix C.1 further shows its faster convergence and superior performance through accuracy and standard deviation curves.

5.3 Ablation Study

In Table 3, we present an ablation study to evaluate the contribution of the DEAL and ISPU modules in PHP-FL under the *normal* participation pattern. When both modules are disabled, the performance significantly degrades, especially on CIFAR-10, where the accuracy drops to 59.88%.

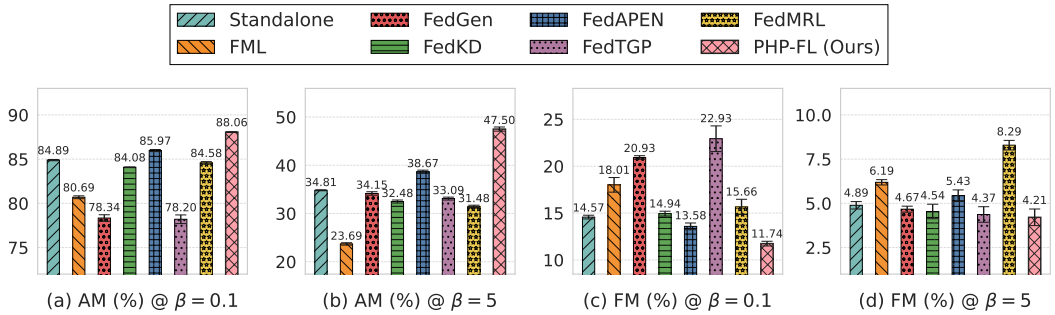
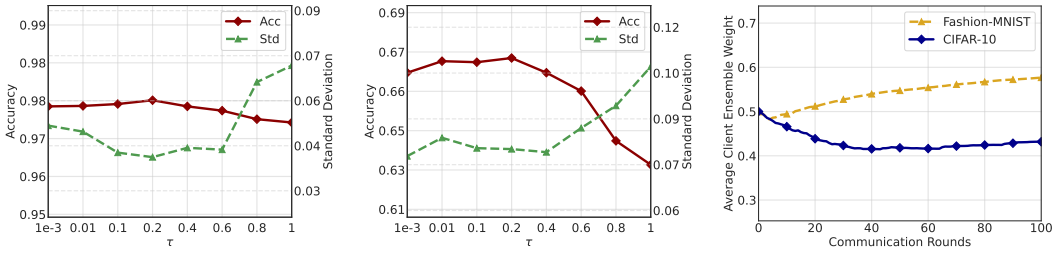


Figure 3: Comparison results on CIFAR-10 under varying degrees of data distribution heterogeneity across clients. All other settings follow their default configurations.



(a) Fashion-MNIST (b) CIFAR-10 (c) Average weight λ

Figure 4: Left two: effect of τ on performance. Right: average weight λ of clients in each round.

Introducing the ISPU module alone brings a modest improvement, highlighting its effectiveness in mitigating update conflicts from inconsistent client participation.

Besides, enabling only the DEAL module yields a more pronounced performance gains. This is achieved by effectively addressing system heterogeneity through data-free knowledge alignment and ensemble learning. Notably, enabling both DEAL and ISPU achieves the best performance on both datasets, with 97.59% accuracy on Fashion-MNIST and 66.94% on CIFAR-10, demonstrating their complementarity and the necessity of their joint design.

Table 3: **Ablation study** of key modules of PHP-FL under the *normal* pattern.

DEAL	ISPU	Fashion-MNIST		CIFAR-10	
		AM (%)	FM (%)	AM (%)	FM (%)
✗	✗	92.86	9.10	59.88	11.24
✗	✓	96.72	5.07	62.05	9.47
✓	✗	96.98	5.73	63.08	8.16
✓	✓	**97.59**	**4.24**	**66.94**	**8.07**

5.4 Case Studies

Robustness to Non-IIDness. To evaluate PHP-FL's robustness under varying data heterogeneity, we conducted additional experiments on CIFAR-10 using the Dirichlet distribution with $\beta = 0.1$ (high heterogeneity) and $\beta = 5$ (low heterogeneity). As shown in Figure 3, PHP-FL consistently outperforms all baselines across both settings. Under high heterogeneity, PHP-FL surpasses the best-performing baseline by 2.09% in accuracy (AM) and reduces the fairness metric (FM) by 1.84%. This advantage becomes even more pronounced under low heterogeneity, where PHP-FL achieves a remarkable 8.83% accuracy gain over the next best method while maintaining the best fairness performance. These results clearly show that PHP-FL is not only robust to different levels of data heterogeneity but consistently achieves state-of-the-art performance in both accuracy and fairness.

Effect of the Pruning Threshold τ on Performance. To investigate the effect of the hyperparameter τ, we conduct experiments on Fashion-MNIST and CIFAR-10 under the *normal* pattern. As shown in Figure 4a and 4b, on the CIFAR-10 dataset, the accuracy first increases and then decreases, while the standard deviation initially decreases and then increases, with both metrics achieving their best values when τ is set to 0.2. For the Fashion-MNIST dataset, the performance remains relatively

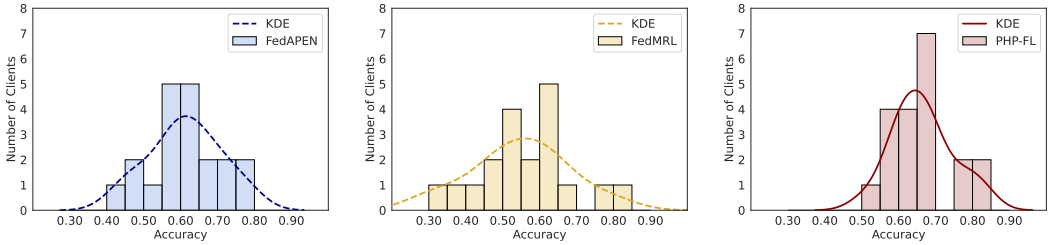


Figure 5: The client accuracy distribution at the best achieved AM (best mean accuracy) on CIFAR-10 dataset under the *uniform* participation pattern for PHP-FL and two other baseline methods.

stable across different τ, and similarly, the best results are also observed at $\tau = 0.2$. Therefore, we choose $\tau = 0.2$ as the default configuration for all experiments.

Effect of Adaptive Ensemble Weights. We analyze the behavior of the adaptive ensemble weight λ by tracking its average value across clients throughout training under the *uniform* pattern. As depicted in Figure 4c, the dynamics of λ differ significantly between datasets. For CIFAR-10, the average λ is initialized as 0.5 but quickly decreases and stabilizes around 0.43, indicating a consistent preference for the global model $\boldsymbol{g}$ within the ensemble on this more complex dataset. In contrast, on Fashion-MNIST, the average λ steadily increases from 0.5 to approximately 0.58 by the end of training, signifying a growing reliance on the specialized local models $\boldsymbol{w}$. This demonstrates that the adaptive mechanism effectively captures dataset-specific characteristics, dynamically adjusting the ensemble balance between local and global models to leverage their respective strengths during the learning process.

Visualization of Client Accuracy Distribution. To visualize the client accuracy distribution under the *normal* participation pattern, we plot histograms and Kernel Density Estimation (KDE) [66] curves for different methods on CIFAR-10 dataset. As shown in Figure 5, PHP-FL achieves a more concentrated accuracy distribution compared to FedAPEN and FedMRL, with clients generally attaining higher accuracy. Moreover, the differences in client performance are significantly reduced under PHP-FL, highlighting its superiority in both enhancing average performance and promoting fairness across clients.

6 Conclusion

In this paper, we propose PHP-FL, a novel model-heterogeneous federated learning method addressing the critical challenge of enhancing both accuracy and fairness under inconsistent client participation probabilities. PHP-FL achieves this through two integrated modules: (1) the DEAL module, which harmonizes heterogeneous models via data-free knowledge alignment; and (2) the ISPU module, which selectively updates task-relevant parameters to mitigate update conflicts. Evaluated across diverse participation patterns, PHP-FL demonstrates state-of-the-art performance for both accuracy and fairness. Ablation study further validates the effectiveness of each module. Our research narrows the divide between idealized uniform participation scenarios and practical heterogeneous FL systems, providing a lightweight yet robust solution suitable for real-world implementation.

Limitations. Despite the promising results, PHP-FL has two main limitations:

First, compared to approaches that exchange only lightweight information (*e.g.*, logits, prototypes [29, 12], or partial model parameters [10, 11]), our method introduces non-negligible computation and communication overheads.[6] Although employing a smaller auxiliary model can alleviate this burden, the additional costs from ensemble training and selective parameter updates still persist.

Second, our evaluation is conducted on a constrained set of model heterogeneity types, datasets, and client participation patterns. Although PHP-FL demonstrates robust performance within these scenarios, its generalizability should be further verified on more diverse and large-scale benchmarks.

[6]For an analysis of the communication and computation costs, please refer to Appendix C.7.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China under Grant No.2024YFE0204500, and in part by the National Natural Science Foundation of China under Grant No.92267104.

References

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

[2] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[3] Yongzhe Jia, Xuyun Zhang, Hongsheng Hu, Kim-Kwang Raymond Choo, Lianyong Qi, Xiaolong Xu, Amin Beheshti, and Wanchun Dou. Dapperfl: Domain adaptive federated learning with model fusion pruning for edge devices. *Advances in Neural Information Processing Systems*, 37:13099–13123, 2024.

[4] Jingwei Liu, Yating Li, Mengjiao Zhao, Lei Liu, and Neeraj Kumar. Epffl: enhancing privacy and fairness in federated learning for distributed e-healthcare data sharing services. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[5] Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics*, 70(1):2508–2516, 2023.

[6] Shuai Wang, Yexuan Fu, Xiang Li, Yunshi Lan, Ming Gao, et al. Dfrd: Data-free robustness distillation for heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 36:17854–17866, 2023.

[7] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8686–8696, 2023.

[8] Xiuwen Fang and Mang Ye. Noise-robust federated learning with model heterogeneous clients. *IEEE Transactions on Mobile Computing*, 2024.

[9] Siyuan Wu, Hao Tian, Weiran Zhang, Tingtong Zhu, Fuwen Tian, Zhehong Wang, and Wanchun Dou. A heterogeneous federated learning method based on dual teachers knowledge distillation. In *International Conference on Advanced Data Mining and Applications*, pages 192–207. Springer, 2024.

[10] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[11] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

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

[13] Liping Yi, Han Yu, Chao Ren, Gang Wang, Xiaoxiao Li, et al. Federated model heterogeneous matryoshka representation learning. *Advances in Neural Information Processing Systems*, 37:66431–66454, 2024.

[14] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[15] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8432–8440, 2022.

[16] Jianqiao Zhang, Caifeng Shan, and Jungong Han. Fedgmkd: An efficient prototype federated learning framework through knowledge distillation and discrepancy-aware aggregation. *Advances in Neural Information Processing Systems*, 37:118326–118356, 2024.

[17] Michal Yemini, Rajarshi Saha, Emre Ozfatura, Deniz Gündüz, and Andrea J Goldsmith. Robust federated learning with connectivity failures: A semi-decentralized framework with collaborative relaying. *arXiv preprint arXiv:2202.11850*, 2022.

[18] Hao Ye, Le Liang, and Geoffrey Ye Li. Decentralized federated learning with unreliable communications. *IEEE journal of selected topics in signal processing*, 16(3):487–500, 2022.

[19] Ming Wen, Chengchang Liu, and Yuedong Xu. Communication efficient distributed newton method over unreliable networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15832–15840, 2024.

[20] Ming Xiang, Stratis Ioannidis, Edmund Yeh, Carlee Joe-Wong, and Lili Su. Efficient federated learning against heterogeneous and non-stationary client unavailability. *Advances in Neural Information Processing Systems*, 37:104281–104328, 2024.

[21] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[22] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR, 2022.

[23] Tiansheng Huang, Weiwei Lin, Wentai Wu, Ligang He, Keqin Li, and Albert Y Zomaya. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1552–1564, 2020.

[24] Rituparna Saha, Sudip Misra, Aishwariya Chakraborty, Chandranath Chatterjee, and Pallav Kumar Deb. Data-centric client selection for federated learning over distributed edge networks. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):675–686, 2023.

[25] Zhiyuan Ning, Chunlin Tian, Meng Xiao, Wei Fan, Pengyang Wang, Li Li, Pengfei Wang, and Yuanchun Zhou. Fedgcs: A generative framework for efficient client selection in federated learning via gradient-based optimization. *arXiv preprint arXiv:2405.06312*, 2024.

[26] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[27] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[28] Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12077–12086, 2024.

[29] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[30] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[31] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10143–10153, 2022.

[32] Jaehee Jang, Heoneok Ha, Dahuin Jung, and Sungroh Yoon. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proceedings of the 51st International Conference on Parallel Processing*, pages 1–10, 2022.

[33] Guangyu Sun, Matias Mendieta, Jun Luo, Shandong Wu, and Chen Chen. Fedperfix: Towards partial model personalization of vision transformers in federated learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4988–4998, 2023.

[34] Feijie Wu, Xingchen Wang, Yaqing Wang, Tianci Liu, Lu Su, and Jing Gao. FIARSE: Model-heterogeneous federated learning via importance-aware submodel extraction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[35] Jingwen Zhang, Yuezhou Wu, and Rong Pan. Incentive mechanism for horizontal federated learning based on reputation and reverse auction. In *Proceedings of the Web Conference 2021*, pages 947–956, 2021.

[36] Meirui Jiang, Holger R Roth, Wenqi Li, Dong Yang, Can Zhao, Vishwesh Nath, Daguang Xu, Qi Dou, and Ziyue Xu. Fair federated medical image segmentation via client contribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16302–16311, 2023.

[37] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[38] Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horvath, and Karthik Nandakumar. Redefining contributions: Shapley-driven federated learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

[39] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7494–7502, 2023.

[40] Zeqing He, Zhibo Wang, Xiaowei Dong, Peng Sun, Ju Ren, and Kui Ren. Towards fair federated learning via unbiased feature aggregation. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[41] Zheng Wang, Xiaoliang Fan, Jianzhong Qi, Chenglu Wen, Cheng Wang, and Rongshan Yu. Federated learning with fair averaging. *arXiv preprint arXiv:2104.14937*, 2021.

[42] Mingyang Wan, Daochen Zha, Ninghao Liu, and Na Zou. In-processing modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, 17(3):1–27, 2023.

[43] Reuben Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 514–524, 2020.

[44] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. *Advances in Neural Information Processing Systems*, 34:12052–12064, 2021.

[45] Shiqiang Wang and Mingyue Ji. A lightweight method for tackling unknown participation statistics in federated averaging. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[47] Zhen Qin, Shuiguang Deng, Mingyu Zhao, and Xueqiang Yan. Fedapen: Personalized cross-silo federated learning with adaptability to statistical heterogeneity. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1954–1964, 2023.

[48] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[49] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

[50] Xiaopeng Jiang and Cristian Borcea. Complement sparsification: Low-overhead model pruning for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8087–8095, 2023.

[51] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group fisher pruning for practical network compression. In *International Conference on Machine Learning*, pages 7021–7032. PMLR, 2021.

[52] Xiyuan Yang, Wenke Huang, and Mang Ye. Fedas: Bridging inconsistency in personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11986–11995, 2024.

[53] Michael Santacroce, Zixin Wen, Yelong Shen, and Yuanzhi Li. What matters in the structured pruning of generative language models? *arXiv preprint arXiv:2302.03773*, 2023.

[54] Xinghao Wu, Xuefeng Liu, Jianwei Niu, Guogang Zhu, and Shaojie Tang. Bold but cautious: Unlocking the potential of personalized federated learning through cautiously aggressive collaboration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19375–19384, 2023.

[55] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, 2017.

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

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

[58] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

[59] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, Jian Cao, and Haibing Guan. Gpfl: Simultaneously learning global and personalized feature information for personalized federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5041–5051, 2023.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[61] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[62] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[64] 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. *Journal of Machine Learning Research*, 26(50):1–10, 2025.

[65] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[66] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[67] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[68] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction Sections accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion about the limitations of this work is provided in the Conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper presents comprehensive details of the experimental setup, and further provides the hyperparameters of baseline methods. This paper also provides an URL link to our released code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: This paper provides an URL link to our released code. The public datasets used in this paper are properly referenced.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper presents comprehensive details of the experimental setup in the Experimental Section and further provides the hyperparameters of baseline methods in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports experimental results averaged over 3 runs with corresponding standard deviations in the main results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper describes the experimental environment in the Experiments Section and our code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper fully complies with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets used in this paper are properly credited. The license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

APPENDIX

A Pseudo codes of PHP-FL

The algorithm is outlined in Algorithm 1. Please refer to Section 4.1 for more details.

Algorithm 1: PHP-FL

Input: Auxiliary small models $\{\boldsymbol{g}_i^0\}_{i=1}^K$, Local models $\{\boldsymbol{w}_i^0\}_{i=1}^K$, Total number of rounds T, Dataset partitions $\{D_i\}_{i=1}^K$.
Output: Optimized local models $\{\boldsymbol{w}_i^T\}_{i=1}^K$ and auxiliary small models $\{\boldsymbol{g}_i^T\}_{i=1}^K$.
Initialize historical mask $\{M_i^h\}_{i=1}^K \leftarrow \mathbf{1}^{|\boldsymbol{g}|}$ and global model $\mathcal{G}^0$;
for *round* $t = 1$ *to* T **do**
  Server Side:
    Collect the IDs of active clients $\mathcal{A}_t \subseteq \{1, ..., K\}$;
    Broadcast the pruned auxiliary small model $\mathcal{G}^{t-1} \odot \boldsymbol{M}_i^h$ to client $i \in \mathcal{A}_t$;
    $\{\boldsymbol{g}_i^t, M_i^t\}_{i \in \mathcal{A}_t} \leftarrow$ **Client Update**;
    Update the historical mask $M_i^h \in M_{\text{hist}}$ for clients in $\mathcal{A}_t$: $M_i^h \leftarrow M_i^t$.
    Aggregate auxiliary small models: $\mathcal{G}^t = \frac{1}{|\mathcal{A}_t|} \sum_{i \in \mathcal{A}_t} \boldsymbol{g}_i^t$;
  Client Update:
  for *each client* $i \in \mathcal{A}_t$ ***in parallel*** **do**
      Download $\mathcal{G}^{t-1} \odot M_i^h$ from server;
      $\hat{\boldsymbol{g}}_i^{t-1} \leftarrow$ initialize local auxiliary model by Eq. 11;
      Randomly divides the training set D_i into D_i^a and D_i^s;
      for *batch* $(x, y) \in D_i^a$ **do**
        $\boldsymbol{\lambda}_i^t \leftarrow$ update ensemble weight by Eq. 5;
      end
      for *batch* $(x, y) \in D_i^s$ **do**
        $\mathcal{L}_{\boldsymbol{w}} \leftarrow$Calculate local training loss by Eq. 6;
        $\mathcal{L}_{\boldsymbol{g}} \leftarrow$ Calculate the symmetrical loss similar to Eq. 6;
        $\boldsymbol{w}_i^t, \boldsymbol{g}_i^t \leftarrow$ update the local model and local auxiliary model by Eq. 7;
      end
      $\alpha_i^t \leftarrow$ calculate update ratio by Eq. 9;
      $M_i^t \leftarrow$ obtain binary mask by Eq. 10 ;
      Upload $\boldsymbol{g}_i^t$ and the binary mask M_i^t to server;
  end
end

B Additional Experimental Details

B.1 Hyperparameter Settings

We provide a detailed summary of the hyperparameter configurations used in our experiments in Table 4. These settings are carefully selected to ensure fair comparison across different baselines. [7] For the proposed PHP-FL, the standardized feature dimension d is set to 512. The adaptability set D_i^a consists of 10% randomly sampled data from the training set D_i. The ensemble weight $\boldsymbol{\lambda}_i^t$ is trained for 10 epochs in each round. Additionally, following the hyperparameter search detailed in Section 5.4 and C.6, the pruning threshold τ and the sharpness factor δ are set to 0.2 and 5, respectively. Unless otherwise specified, all experiments follow the same training setting.

[7] Note that these parameter names in different methods are consistent with the original references and are independent of the notation used in our work.

Table 4: Hyperparameter settings used in our experiments.

Type	**Hyperparameters**	**Value**
FL training settings	Local learning rate η	0.001
	Batch size	64
	Local epochs per round E	5
	Total rounds T	100
	Number of clients K	20
Framework-specific	α in FML	0.5
	β in FML	0.5
	Server learning epochs in FedGen	100
	Server learning rate in FedGen	0.1
	d_η in FedGen	32
	d_h in FedGen	512
	T_{start} in FedKD	0.95
	T_{end} in FedKD	0.98
	η_s in FedKD	0.001
	Adaptation set ratio in FedAPEN	10%
	Server learning epochs in FedTGP	100
	τ in FedTGP	100
	λ in FedTGP	0.1
	d_1 in FedMRL	128
	λ in Ditto	0.1
	α for CIFAR-10 in FedFV	0.1
	α for Fashion-MNIST in FedFV	0
	τ for CIFAR-10 in FedFV	10
	τ for Fashion-MNIST in FedFV	0
	β in FedHEAL	0.4
	τ in FedHEAL	0.4
	K in FedAU	1

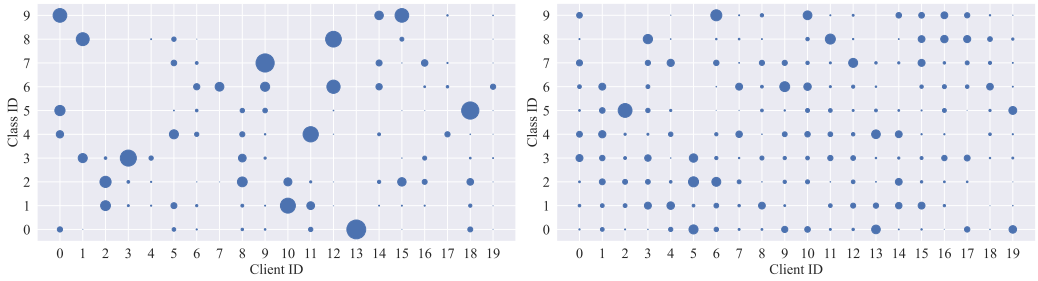
B.2 Visualization of Data Distributions

To intuitively illustrate the data heterogeneity across clients in our federated learning setting, we plot scatter diagrams based on the CIFAR-10 and Fashion-MNIST datasets in Figure 6. Specifically, we simulate heterogeneous data distributions by allocating the proportion of class j to each client k according to a Dirichlet distribution ($p_{j,k} \sim \text{Dir}(\beta)$), where a smaller β indicates more extreme data heterogeneity across clients. In our experiments, we set $\beta = 0.1$ for Fashion-MNIST and $\beta = 0.5$ for CIFAR-10, respectively.

B.3 Model Architectures Used in Experiments

we utilize four widely recognized Convolutional Neural Network (CNN) architectures with varying designs and complexities. We report the corresponding parameter counts of each model in Table 5. These serve as the local models for clients in our heterogeneous federated learning setup:

- **GoogLeNet** [60]: Introduced the inception module, which performs convolutions with multiple filter sizes in parallel within the same block. It was designed for computational efficiency and won the ILSVRC 2014 challenge.
- **DenseNet-121** [61]: Characterized by its dense connectivity pattern, where each layer receives feature maps from all preceding layers within a dense block. This encourages feature reuse, strengthens gradient flow, and improves parameter efficiency. The '121' denotes the number of layers with weights.
- **EfficientNet-B1** [62]: Developed using neural architecture search and a compound scaling method that uniformly scales network width, depth, and resolution. It aims to balance model accuracy with computational efficiency (FLOPS and parameters). B1 is a specific scaled version providing a good trade-off.



(a) Fashion-MNIST (b) CIFAR-10

Figure 6: The data distribution of 20 clients in our experiments.

- **ResNet-18** [63]: Employs residual connections (skip connections) that allow gradients to bypass layers, enabling the training of much deeper networks by mitigating the vanishing gradient problem. ResNet-18 is a relatively shallow variant with 18 layers containing weights.

For a fair comparison, we adopt GoogLeNet [60] as the auxiliary model architecture for all methods that utilize an auxiliary model, including our proposed method PHP-FL, since it has the smallest number of parameters among the four candidate models.

Table 5: Parameter counts of the evaluated models. "M" is short for million.

Model	Parameter counts
GoogleNet [60]	5.61M
DenseNet-121 [61]	6.96M
EfficientNet-B1 [62]	6.52M
ResNet-18 [63]	11.18M

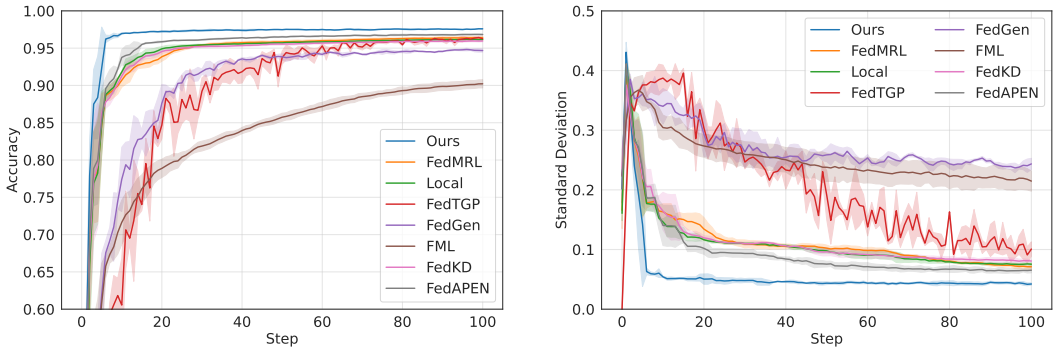
C Additional Experimental Results

C.1 Comparison of Training Curves for Accuracy and Standard Deviation

To evaluate the convergence behavior and training stability of PHP-FL, we compare the training curves in terms of average accuracy and standard deviation across training rounds on Fashion-MNIST and CIFAR-10 under the *uniform* pattern. As shown in Figure 7 and 8, the results on both datasets demonstrate that the proposed method PHP-FL significantly accelerates convergence compared to baselines, while maintaining a low and stable standard deviation throughout training, which effectively enhances performance and fairness across clients.

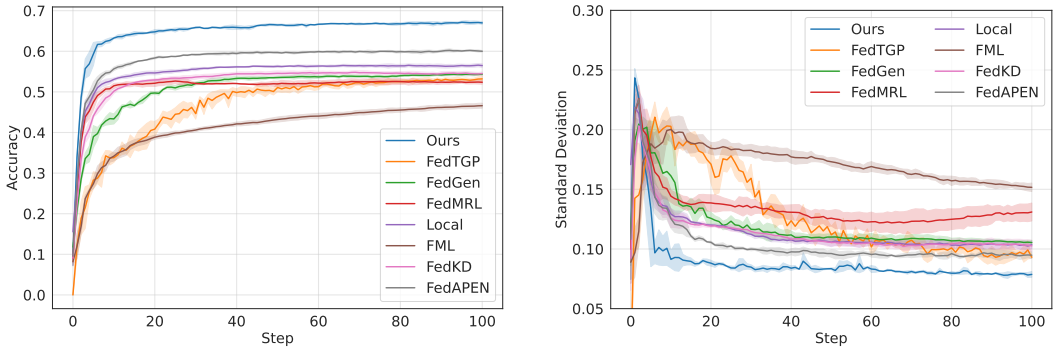
C.2 Comparison to More State-of-the-Arts Methods in the Homogeneous Setting

To further validate the generality and effectiveness of PHP-FL, we conduct additional experiments in the homogeneous setting and report the results in Table 6. Specifically, all clients adopt the identical GoogLeNet architectures [60]. Additional compared methods include FedFV [41] and FedHEAL [28], which are designed to improve client fairness (**Fair-FL**), as well as FedAWE [20] and FedAU [45], which address client unavailability (**CU-FL**). As shown in Table 6, PHP-FL achieves the highest average accuracy (AM) across all patterns, while also maintaining a highly competitive performance fairness (FM). Although FedHEAL exhibits slightly better fairness, its average accuracy is significantly inferior compared to PHP-FL. This highlights that PHP-FL not only delivers the highest average performance but also achieves fairness that is highly competitive with the best fair-FL methods, striking an exceptional balance that surpasses existing methods in overall effectiveness even in the homogeneous setting. These results underscore the robustness and practicality of PHP-FL even in homogeneous FL scenarios, reaffirming its overall superiority.



(a) Average accuracy (b) Standard deviation

Figure 7: Comparison of training curves on Fashion-MNIST.



(a) Average accuracy (b) Standard deviation

Figure 8: Comparison of training curves on CIFAR-10.

Table 6: Comparison with the state-of-the-art methods on CIFAR-10 in the homogeneous setting. Best in **bold** and second with underline.

Type	Methods	***Uniform*** $[a = 0.5]$		***Normal*** $[\mu = 0.5, \sigma = 0.2]$		***Linear*** $\left[a = 0.05, d = \frac{K-2}{K(K-1)}\right]$	
		AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓
MH-FL	FML [arXiv20]	41.31±0.46	13.28±0.31	41.00±0.48	13.46±0.35	39.21±2.80	13.37±0.30
	FedGen [ICML21]	61.79±0.13	8.80±0.46	61.73±0.15	8.95±0.25	61.60±0.29	8.99±0.20
	FedKD [NC22]	61.26±0.08	10.03±0.10	60.96±0.38	9.93±0.11	60.73±0.70	10.01±0.09
	FedAPEN [KDD23]	66.49±0.30	8.14±0.16	66.54±0.31	8.26±0.18	66.60±0.35	8.25±0.17
	FedTGP [AAAI24]	60.12±0.48	9.97±0.46	60.15±0.46	10.20±0.31	59.88±0.69	10.14±0.31
	FedMRL [NIPS24]	62.86±0.62	8.60±0.16	62.86±0.62	8.95±0.58	62.58±0.54	8.80±0.38
Fair-FL	FedFV [IJCAI21]	59.75±0.87	11.03±1.63	60.05±1.31	10.54±2.02	58.32±1.45	11.08±1.60
	FedHEAL [CVPR24]	59.43±0.83	**7.03**±0.76	59.36±0.75	**7.93±1.14**	59.15±0.57	**7.94±1.15**
CU-FL	FedAWE [NIPS24]	62.75±0.09	9.39±0.30	62.68±0.13	9.31±0.25	62.54±0.31	9.36±0.28
	FedAU [ICLR24]	62.79±0.20	9.17±0.19	62.72±0.25	9.21±0.29	62.59±0.47	9.35±0.15
All	**PHP-FL (Ours)**	**67.99±0.16**	8.05±0.19	**67.88±0.32**	8.08±0.16	**67.84±0.13**	8.12±0.12

C.3 Performance with Varying Numbers of Clients

In this Section, we compare the performance of different methods with varying numbers of clients on CIFAR-10. Specifically, the *uniform* pattern involves 10 clients with full participation in every round; the *normal* pattern consists of 50 clients whose participation probabilities are sampled from a normal distribution ($\mu = 0.2$, $\sigma = 0.2$), resulting in an average of 10 clients per round; and the *linear* pattern also includes 50 clients, with participation probabilities increasing linearly from 0.02 by a step of $\frac{0.36}{K-1}$, yielding the same average of 10 clients per round. As shown in Table 7, PHP-FL consistently achieves the best accuracy (AM) and performance fairness (FM) across all three patterns. Notably, compared to the strongest baseline FedAPEN, our method improves AM by 6.62%, 6.47%, and 6.12% under the *uniform*, *normal*, and *linear* settings, respectively, while also reducing FM, demonstrating superior robustness and fairness with varying numbers of clients.

Table 7: Comparison with the state-of-the-art methods with varying numbers of clients on CIFAR-10 in the heterogeneous setting. Best in **bold** and second with underline.

Methods	***Uniform*** $[a = 1.0]$		***Normal*** $[\mu = 0.2, \sigma = 0.2]$		***Linear*** $\left[a = 0.02, d = \frac{0.36}{K-1}\right]$	
	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓
Standalone	58.89±0.20	10.53±0.24	50.73±1.29	14.04±1.40	51.81±0.30	12.81±0.47
FML [arXiv20]	46.61±0.61	15.18±0.34	36.59±1.17	17.35±1.22	35.26±0.22	17.10±0.16
FedGen [ICML21]	57.82±0.06	9.45±0.29	43.70±1.03	15.10±1.26	43.35±1.08	14.92±0.53
FedKD [NC22]	57.46±0.45	11.26±0.33	49.27±0.93	13.98±0.63	49.07±0.62	13.99±0.71
FedAPEN [KDD23]	63.90±0.10	9.05±0.51	52.99±1.23	13.68±1.77	53.76±0.39	12.74±0.27
FedTGP [AAAI24]	58.45±0.69	9.88±0.37	40.15±0.32	14.74±0.69	38.18±1.04	16.08±1.48
FedMRL [NIPS24]	58.33±0.20	14.23±0.63	49.32±1.34	13.64±1.01	49.26±0.74	13.26±0.27
PHP-FL (Ours)	**70.52±0.03**	**7.97±0.49**	**59.46±0.83**	**12.69±1.41**	**59.88±0.26**	**11.22±0.37**

Table 8: Comparison results on CIFAR-10 dataset under the *Markovian* participation pattern. All other settings follow their default configurations.

Method	Standalone	FML	FedGen	FedKD	FedAPEN	FedTGP	FedMRL	PHP-FL (Ours)
AM (%) ↑	52.77±1.44	42.73±0.96	51.93±0.84	51.31±0.80	57.84±0.53	50.55±1.79	49.02±1.74	**64.04±0.73**
FM (%) ↓	16.10±2.50	19.29±2.04	14.91±0.38	16.82±1.45	13.99±3.34	12.14±1.66	15.86±0.08	**11.69±1.35**

C.4 Performance under the *Markovian* Participation Pattern

To better reflect dynamic client availability in real-word scenarios, we have conducted additional experiments under the *Markovian* participation pattern following FedAU [45]. In this pattern, each client follows a two-state Markov chain to determine its participation status in each training round, where the two states correspond to "participating" and "not participating" in the current training round. This modeling introduces temporal correlation in client participation behavior while maintaining sufficient randomness, offering a more realistic simulation compared to independently sampled participation patterns.

For the parameter settings of the *Markovian* pattern experiment, we constrain the maximum transition probability from the non-participating state to the participating state to 0.05, thereby avoiding excessively frequent state changes and ensuring realistic participation dynamics. The initial state of each client's Markov chain is randomly sampled according to the stationary probability, which is set to 0.5 to ensure that approximately half of the clients participate in each round on average. The transition probabilities are carefully calibrated to maintain this stationary distribution throughout the training process, ensuring system stability while introducing participation heterogeneity across clients. As shown in Table 8, the experimental results under the *Markovian* participation pattern further validate PHP-FLs robustness, consistently achieving superior accuracy and fairness despite the increased dynamic and unpredictable client availability.

C.5 Effectiveness of Adaptive Selective Updates in the ISPU Module

To further validate the design rationality of the adaptive selective parameter update mechanism in our ISPU module, we conduct an ablation study comparing different update strategies for the local auxiliary model $\boldsymbol{g}$, focusing on the update ratio α. Specifically, we evaluate the following variants:

- **Fixed update.** Updates a fixed ratio of the most important parameters, with α set to a constant and we adopt $\alpha = 0.5$.
- **Full Update.** Updates all parameters without selection, which is equivalent to $\alpha = 1$.
- **Random Update.** All parameters are stochastically updated with probability α, where α is dynamically determined by our proposed method using Eq. 9.
- **Adaptive update (Ours).** Dynamically adjusts α using Eq. 9 based on client participation history and parameter importance.

As shown in Table 9, our adaptive update mechanism consistently achieves the best accuracy and performance fairness. Full update shows the poorest performance, and while fixed Update offers some stability, it is surpassed by the adaptive methods. Random update achieves competitive accuracy but inferior fairness compared to our method. The findings highlight the superiority of our adaptive selective update mechanism within the ISPU module.

Table 9: Effectiveness of Adaptive Selective Updates in the ISPU module. Results are evaluated under the *linear* participation pattern on Fashion-MNIST and CIFAR-10. Best in **bold**.

Different variants	Fashion-MNIST		CIFAR-10	
	AM (%) ↑	FM (%) ↓	AM (%) ↑	FM (%) ↓
Fixed Update ($\alpha = 0.5$)	97.43	4.41	64.11	9.14
Full Update ($\alpha = 1.0$)	96.82	5.73	62.81	8.65
Random Update (dynamic α)	97.28	6.23	65.97	8.97
Adaptive Update (**Ours** with dynamic α)	**97.58**	**4.29**	**66.78**	**8.09**

C.6 Effect of the Sharpness Factor on Performance

In our proposed PHP-FL, the hyperparameter δ in Eq. 9 controls the sharpness of the mapping from local participation frequency to the update ratio α in the ISPU module. A larger δ causes α to approach 1 for frequently participating clients and 0 for infrequent ones, whereas a smaller δ smooths the adjustment, pushing α toward 0.5. As shown in Table 10, performance is relatively stable across a range of δ values, but we observe that $\delta = 5$ consistently achieves near-optimal results in both accuracy and fairness across Fashion-MNIST and CIFAR-10. Therefore, we set $\delta = 5$ in our experimental configuration.

Table 10: Effect of the hyperparameter δ on performance. Best in **bold** and second with underline.

Dataset	Metric	$\delta = 0.1$	$\delta = 0.5$	$\delta = 1$	$\delta = 5$	$\delta = 10$	$\delta = 50$	$\delta = 100$
Fashion-MNIST	AM (%) ↑	97.54	**97.62**	97.59	<u>97.61</u>	97.59	97.52	97.51
	FM (%) ↓	4.25	<u>4.13</u>	4.39	**4.12**	4.60	4.35	4.43
CIFAR-10	AM (%) ↑	67.24	**67.28**	67.06	<u>67.27</u>	66.36	67.17	66.82
	FM (%) ↓	8.44	<u>8.10</u>	8.12	**8.01**	8.51	8.49	8.55

C.7 Cost and Efficiency Analysis

Communication Cost. We compare the per-round communication cost with baselines in terms of the number of parameters transmitted between 20 clients and the server on CIFAR-10. As shown in Table 11, among the evaluated baselines, methods such as FedGen and FedTGP demonstrate significantly lower communication costs due to their use of partial model sharing and lightweight

prototypes. Unfortunately, methods relying on auxiliary model transmission (*e.g.*, FML, FedKD, FedAPEN, and FedMRL) exhibit communication cost exceeding 200M parameters per round. In contrast, PHP-FL requires only 112.52M parameters per round, which is nearly half the cost of other auxiliary model-based approaches such as FedAPEN and FedMRL. This reduction is primarily due to clients downloading only the pruned global auxiliary model parameters instead of the full model.

Computation Cost. We also report the total computation cost per round across all clients in terms of FLOPs (floating-point operations),[8] as summarized in Table 11. Following [12], other operations such as data preprocessing are not included in the FLOPs calculation. To ensure a fair comparison, this experiment involves 20 clients with full participation in each round. All other configurations remain aligned with the main experiments.

According to Table 11, PHP-FL incurs the per-round computation cost at 813.71GFLOPs. This increase is marginal when compared with other auxiliary model-based methods such as FML (753.51G), FedKD (753.51G), FedAPEN (771.01G), and FedMRL (757.34G). Compared to FedGen (391.38G) and FedTGP (387.99G), the increased computation cost primarily arises from the training of additional auxiliary models. Besides, the slightly higher cost of the proposed PHP-FL is mainly attributed to the extra training required for ensemble weights. While FedTGP achieves the lowest computation cost, its accuracy significantly lags. Despite this slight increase in cost compared with other auxiliary model-based methods, PHP-FL achieves significantly the best results in both accuracy and performance fairness, as shown in the main experiments. The results underscore a compelling trade-off, with our method delivering notable performance gains at the cost of only a slight increase in computation.

Table 11: Comparison of communication and computation costs per round on CIFAR-10, where communication cost is measured by the number of parameters transmitted between 20 clients and the server, and computation cost is evaluated as the total number of FLOPs executed across all 20 clients. 'M' and 'G' denote million and giga, respectively. For FedKD, the SVD computation cost is excluded from this analysis.

Method	Communication Cost	Computation Cost
FML [arXiv20]	224.20M	753.51G
FedGen [ICML21]	5.92M	391.38G
FedKD [NC22]	200.26M	753.51G
FedAPEN [KDD23]	224.20M	771.01G
FedTGP [AAAI24]	0.192M	387.99G
FedMRL [NeurIPS24]	224.05M	757.34G
PHP-FL (Ours)	112.52M	813.71G

Efficiency Analysis. In addition, we conducted further experiments to measure the total costs (communication rounds, computation cost, and communication cost) required to reach 50% accuracy compared to baselines. As shown in Figure 9, while PHP-FL's per-round communication cost is higher than non auxiliary model-based methods (*e.g.*, FedGen and FedTGP) due to the transmission of auxiliary model, it reaches target accuracy in just 2 rounds, whereas all baselines require at least 5 rounds. Consequently, the total communication and computation costs are substantially lower. Furthermore, the per-round costs can be readily optimized in practice by employing smaller auxiliary models or mask quantization (*e.g.*, bitwidth reduction). Thus, PHP-FL offers a far more efficient path to convergence in practical deployments.

D Discussion

Communication Cost. In the auxiliary model-based methods, active clients often need to upload $L = |\boldsymbol{g}|$ parameters of the global auxiliary model in each communication round, which are typically represented in full-precision floating-point format. Our method similarly requires uploading

[8]We calculate FLOPs using the thop library: `https://github.com/Lyken17/pytorch-OpCounter`.

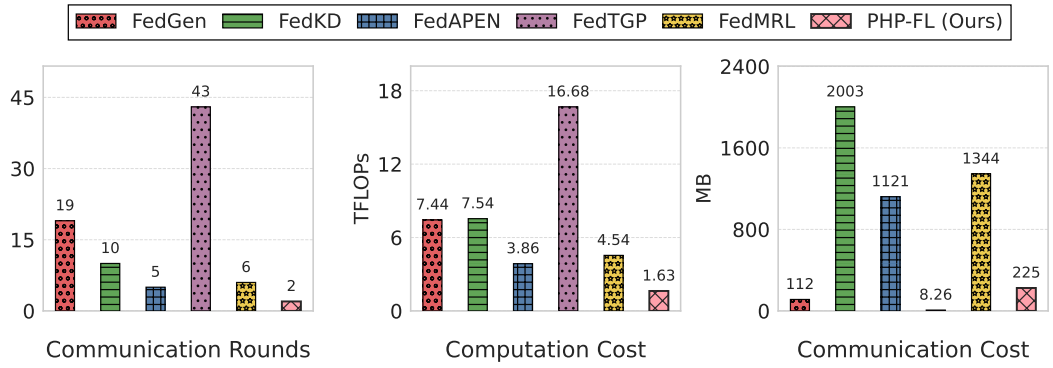


Figure 9: From left to right: Communication rounds, total number of transmitted parameters, and computation FLOPs required to achieve the 50% accuracy on CIFAR-10 under the *uniform* pattern.

the global auxiliary model but with an additional binary mask matrix M of the same dimension L. Fortunately, since each element in M is a binary value (0 or 1), it only incues **1 bit per parameter**, rendering the added communication overhead negligible compared to the transmission of full-precision parameters. Moreover, when downloading the global auxiliary model, clients only need to receive $\alpha \cdot L$ ($\alpha \in (0, 1)$) parameters, where the pruning threshold τ in Eq 9 can be adjusted as needed. This flexibility allows us to further reduce communication costs dynamically.

Participation Patterns. We clearly state that we make no assumptions about the distribution of client participation patterns and allow them to be arbitrary throughout the training process. Moreover, similar to [45], we do not require any prior knowledge of the client sampling process for the proposed method PHP-FL.

Privacy. Similar to FedAvg [1], PHP-FL does not require sharing raw data or client-specific heterogeneous model parameters. Instead, only the lightweight, homogeneous auxiliary models and the mask matrix related to model parameters are uploaded to the server. This design ensures that sensitive local model structures and data remain on the client side, making PHP-FL suitable for privacy-critical applications. Therefore, PHP-FL is compatible with standard privacy-preserving mechanisms such as secure aggregation [67]. Differentially private variants of FedAvg [68] can be seamlessly integrated similarly.