FedPeWS: Personalized Warmup via Subnetworks for Enhanced Heterogeneous Federated Learning

Nurbek Tastan[1], Samuel Horváth[1], Martin Takáč[1], Karthik Nandakumar[1,2]
[1]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), UAE
[2]Michigan State University (MSU), USA
{nurbek.tastan,samuel.horvath,martin.takac}@mbzuai.ac.ae, nandakum@msu.edu

Statistical data heterogeneity is a significant barrier to convergence in federated learning (FL). While prior work has advanced heterogeneous FL through better optimization objectives, these methods fall short when there is *extreme* data heterogeneity among collaborating participants. We hypothesize that convergence under extreme data heterogeneity is primarily hindered due to the aggregation of conflicting updates from the participants in the initial collaboration rounds. To overcome this problem, we propose a warmup phase where each participant learns a personalized mask and updates only a subnetwork of the full model. This *personalized warmup* allows the participants to focus initially on learning specific *subnetworks* tailored to the heterogeneity of their data. After the warmup phase, the participants revert to standard federated optimization, where all parameters are communicated. We empirically demonstrate that the proposed personalized warmup via subnetworks (`FedPeWS`) approach improves accuracy and convergence speed over standard federated optimization methods. The code can be found at https://github.com/tnurbek/fedpews.

1. Introduction

Federated learning (FL) is a distributed learning paradigm where participants collaboratively train a global model by performing local training on their data and periodically sharing local updates with the server. The server, in turn, aggregates the local updates to obtain the global model, which is then transmitted to the participants for the next round of training [1]. While FL preserves data confidentiality by avoiding collating participant data at the server, *statistical heterogeneity* between local data distributions is a significant challenge in FL [2]. Several attempts have been made to tackle heterogeneity via federated optimization algorithms [3–10], dropout [11, 12], and batch normalization [13].

Consider the scenario where multiple hospitals collaborate to learn a medical image classification model that works across imaging modalities and organs, where the data from each hospital pertains to a different modality (e.g., histopathology, CT, X-ray, ultrasound, etc.) and/or organ (e.g., brain, kidney, colon, etc.). Most of the existing heterogeneous FL algorithms fail when there is such *extreme* data heterogeneity among collaborating participants. The primary reason for this failure is the high degree of conflicts between the local updates and the inherent bias the algorithms introduce, often favoring one modality (dataset) over the others. While enforcing a strong regularization constraint on the local updates [6] can partially alleviate this problem, it dramatically slows down local learning and hence, convergence speed.

In this work, we explore an alternate approach to minimize the initial conflicts between heterogeneous participants by allowing participants in FL to initially train a partial subnetwork using only their local datasets. This warmup phase enables the participants to focus first on learning their local data well before engaging in broader collaboration. Thus, our proposed approach can be summarized as follows (see Figure 1). Initially, each participant uses a personalized binary mask, allowing them to first learn their local data distributions and optimize their local (sparse) models. During this warmup phase, participants transmit only their masked updates to the server, and this process continues for

Second Conference on Parsimony and Learning (CPAL 2025).

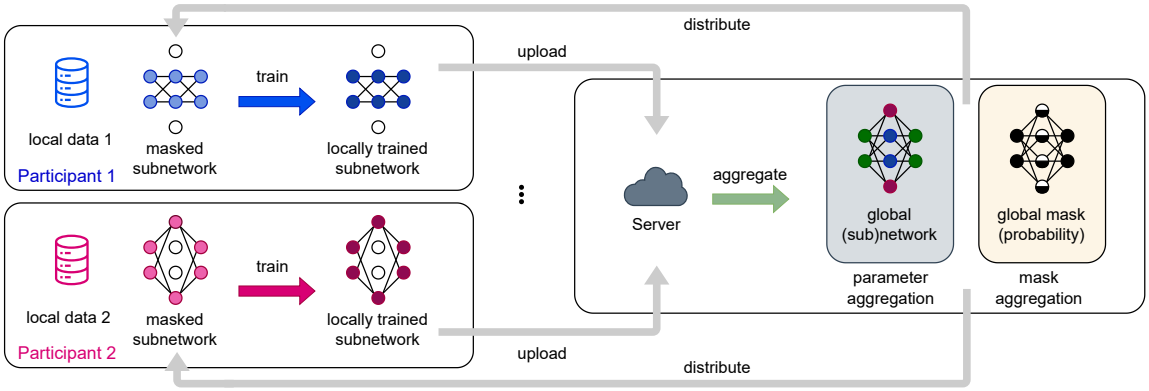


Figure 1: Conceptual illustration of training personalized subnetworks in federated learning.

a certain number of collaboration rounds. At the end of the warmup phase, the participants switch to standard federated optimization for subsequent collaboration. Our contributions are as follows:

1. We introduce a novel concept in federated learning, termed as *personalized warmup via subnetworks* (`FedPeWS`), which helps the global model to generalize to a better solution in fewer communication rounds. This is achieved through a neuron-level personalized masking strategy that is compatible with other FL optimization methods.
2. We propose an algorithm to *identify suitable subnetworks* (subset of neurons) for each participant by simultaneously learning the personalized masks and parameter updates. The proposed algorithm does not make any assumptions regarding the data distributions and incorporates a *mask diversity loss* to improve the coverage of all neurons in the global model.
3. For simple cases involving a small number of participants with known data distributions, we show that it is possible to skip the mask learning step and use fixed masks (that partition the network) determined by the server. We refer to this variant as `FedPeWS-Fixed`.
4. We empirically demonstrate the efficacy of the `FedPeWS` approach under both extreme non-i.i.d. and i.i.d. data scenarios using three datasets: a custom synthetic dataset, a combination of MNIST and CIFAR-10 datasets, a combination of three distinct medical datasets (PathMNIST, OCTMNIST and TissueMNIST), and CIFAR-100 dataset.

2. Related Work

Collaborative Learning. FL is a distributed learning paradigm that addresses data confidentiality concerns [2], particularly in environments where data can not be centralized due to regulatory or practical reasons [14]. One of the seminal FL algorithms, FedAvg [1], involves participants training models locally on their data and periodically transmitting their model parameters to a central server. The server averages these parameters to update the global model, which is then redistributed to the participants for further local refinement. FedAvg has inspired a plethora of variants and extensions aimed at enhancing performance [6, 7, 15], scalability [16, 17], communication efficiency [18–20], privacy/confidentiality [18, 21, 22], robustness [23], and fairness [24–26]. For example, strategies such as weighted averaging or adaptive aggregation have been proposed to accommodate the non-i.i.d. nature of distributed data sources – a scenario where data is not identically distributed across all participants, which can significantly hinder model performance [6, 7, 13, 27–29]. Specifically, FedProx [6] addresses data heterogeneity by integrating a proximal term into the FedAvg framework. There is also a body of work that focuses on addressing the heterogeneity problem through personalization-based approaches, utilizing local-centric objectives [9, 30–34].

Independent Subnet Training. Independent subnet training (IST) is a variant of distributed learning that focuses on enhancing model personalization and reducing communication overhead by training

Table 1: Comparison of approaches for handling data heterogeneity in federated learning.

	Shared Global Model	Level of Mask Personalization	Learnable Mask	Learnable Parameters
FedPM [20]	✓	parameter-level	✓	×
IST [35]	✓	neuron-level	× (random)	✓
LTH [41]	×	parameter-level	× (pruned)	✓
FedWeIT [32]	×	parameter-level	✓	✓
FjORD [11]	✓	parameter-level	× (slimmed)	✓
`FedPeWS` (ours)	✓	neuron-level	✓	✓

separate subnetworks for different participants [35]. IST distributes neurons of a fully connected neural network disjointly across different participants, forming a group of subnets. Then, each of these subnets is trained independently for one or more local SGD steps before synchronization. In every round, after broadcasting the server weights, each participant gets updated neurons to focus on, and the local subnet training continues. This approach led to different works along the line of using subnetwork training for efficiency [11, 12, 36–40] in FL. In our work, we adopt IST's core principle of selecting neurons rather than focusing on weight values, which in turn narrows the search space. A key distinction between our method and IST lies in how the neurons are selected and the necessity of covering all neurons. Whilst IST typically involves random sampling of masks in each training round by the server and full coverage of neurons, we do not randomly sample neurons; instead, we use a learnable mask for each participant that is trained along with the parameters, and we relax the assumption of full coverage of neurons.

Finding Subnetworks in FL. Another relevant idea is the Lottery Ticket Hypothesis (LTH) [41], which attempts to identify subnetworks within a larger network. LTH is a model personalization technique, which focuses on sparsifying the network to create a smaller-scale version that improves per-round communication efficiency. In contrast to LTH, our method is directed towards training a shared global model and simultaneously improving convergence speed (reducing the number of communication rounds). After LTH, there has been a growing interest in finding sparse and trainable networks at initialization [42–44]. In [44, 45], the goal is to learn a personalized model for each client with better communication efficiency. While the sparse subnetworks are learned through pruning in [44], dynamic sparse training-based mask search is employed in [45]. The work by Babakniya et al. [46] focuses on model compression, i.e., learning a sparse server model by identifying a sparse subnetwork within the global model learned via FL. To elaborate further, [44] leverages the lottery ticket hypothesis to find lottery ticket networks inside the main network, and only these networks are communicated during training. The subnetworks are located using pruning techniques, with pruning rates assigned by the server based on hard-coded thresholds for accuracy gains and target pruning rates. At the end of the collaboration, clients obtain personalized models. [45] is a personalized FL technique that employs sparse-to-sparse training with the goal of reducing communication, computation, and memory costs. Sparse models are achieved by applying weight magnitude pruning combined with gradient information computed from a randomly sampled batch. Furthermore, [46] is a sparse learning framework primarily focused on model compression. It identifies subnetworks through sensitivity analysis on a randomly generated mask, which is used to identify a good initial mask for each layer. Our approach fundamentally differs from these methods by aiming to train a single global (dense) model while simultaneously improving the convergence speed (reducing the number of communication rounds required). `FedPeWS` identifies sparse, personalized subnetworks only during the warmup phase of the training to enable quicker/better convergence under extreme non-IID settings.

Recently, in [20], sparse networks were found inside the main model to increase communication efficiency in FL. The proposed FedPM method focuses on finding a subnetwork by freezing the model weights and training for masks on a weight level, in contrast to IST, which works on a neuron level. FedPM utilizes the sigmoid function to obtain probability values from unbounded mask scores

and then uses Bernoulli sampling to obtain binary masks. We use a similar approach in our FedPeWS algorithm to learn the neuron-level personalized masks.

Table 1 presents a comprehensive comparison of our proposed method, FedPeWS, against existing literature for addressing data heterogeneity in federated learning. The comparison highlights key attributes of each method, including the level of mask personalization and whether the masks and parameters are learnable.

Mixture of Experts. Mixture of Experts (MoE) [47, 48] is a technique that trains multiple specialized models (experts) alongside a gating network that dynamically selects which experts to activate for a given input during both training and inference. Although MoE and our proposed method share the concept of leveraging different model components for learning, there is a fundamental distinction. FedPeWS is not designed to train multiple expert models; rather, it learns a single global model while identifying sparse, personalized subnetworks exclusively during the warmup phase of training. This warmup strategy serves as a crucial initialization step to facilitate efficient federated optimization, improving both convergence speed and generalization in heterogeneous FL environments.

3. Preliminaries

Our goal is to minimize a sum-structured federated learning optimization objective:

$$x^\star \leftarrow \arg\min_{x\in\mathbb{R}^d} \left[f(x) \coloneqq \frac{1}{N}\sum_{i=1}^{N} f_i(x) \right], \tag{1}$$

where the components $f_i : \mathbb{R}^d \to \mathbb{R}$ are distributed among N local participants and are expressed in a stochastic format as $f_i(x) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}_i}\big[F_i(x,\xi)\big]$. Here, $\mathcal{D}_i$ represents the distribution of ξ at participant $i \in [N] \coloneqq \{1,\dots,N\}$. This problem encapsulates standard empirical risk minimization as a particular case when each $\mathcal{D}_i$ is represented by a finite set of n_i elements, i.e., $\xi_i = \big\{\xi_i^1,\dots,\xi_i^{n_i}\big\}$. In such cases, f_i simplifies to $f_i(x,\xi_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} F_i(x,\xi_i^j)$. Our approach does not impose restrictive assumptions on the data distribution $\mathcal{D}_i$. In fact, we specifically focus on the extreme heterogeneous (non-i.i.d.) setting, where $\mathcal{D}_i \neq \mathcal{D}_{i'}, \forall\, i \neq i'$ and the *local optimal solution* $x_i^\star \leftarrow \arg\min_{x\in\mathbb{R}^d} f_i(x)$ might significantly differ from the global minimizer of the objective function in Equation 1.

We are especially interested in the supervised classification task and let $\mathcal{M}_x : \mathcal{Z} \to \mathcal{Y}$ be a classifier parameterized by x. Here, $\mathcal{Z} \subseteq \mathbb{R}^D$ and $\mathcal{Y} = \{1,2,\dots,M\}$ denote the input and label spaces, respectively, D is the input dimensionality, M is the number of classes, and d represents the number of parameters in the model $\mathcal{M}$. We set $F_i\left(x,\xi_i^j\right) = \mathcal{L}\left(\mathcal{M}_x\left(\mathbf{z}_i^j\right), y_i^j\right)$, where $\mathcal{L}$ is an appropriate loss function and $\xi_i^j \coloneqq \left(\mathbf{z}_i^j, y_i^j\right)$ is a labeled training sample such that $\mathbf{z}_i^j \in \mathcal{Z}$ and $y_i^j \in \mathcal{Y}$. Furthermore, we mainly focus on the cross-silo FL setting (N is small).

Federated Averaging (FedAvg). A common approach for solving Equation 1 in the distributed setting is FedAvg [1]. This algorithm involves the participants performing K local steps of stochastic gradient descent (SGD) and communicating with the server over T communication rounds. The server initializes the global model with x_g^0 and broadcasts it to all participants, which is then used to initialize the local models, i.e., $x_i^{1,0} = x_g^0$. In each communication round, the updates from the participants are averaged on the server and sent back to all participants. For a local step $k \in [K]$, communication round $t \in [T]$, and participant $i \in [N]$, the local and global iterates are updated as:

$$x_i^{t,k} = x_i^{t,k-1} - \eta_\ell \nabla f_i\left(x_i^{t,k-1}, \xi_i^{t,k-1}\right), \quad x_i^t = x_i^{t,K}, \text{ and } x_g^t = x_g^{t-1} - \eta_g\left(x_g^{t-1} - \frac{1}{N}\sum_{i=1}^{N} x_i^t\right), \tag{2}$$

where η_ℓ and η_g are the local and global learning rates, respectively. The server then broadcasts the updated global model x_g^t to all participants, which is then used to reinitialize the local models as $x_i^{t+1,0} = x_g^t$.

Algorithm 1 `FedPeWS` (For `FedPeWS-Fixed` variant, the steps highlighted in green are omitted and instead the server sets $m_i^t = m_i, \forall\, t \in [W]$.)

Input: Number of collaboration rounds T, number of warmup rounds W, number of local steps K, local learning rate η_ℓ, global learning rate η_g, mask learning rate η_s, λ (mask diversity weight)

Initialize x_g^0 and s_g^0, compute $\theta_g^0 = \sigma(s_g^0)$
for $t = 1, \ldots, T$ **do**
  if $t > W$ **then** // Use all parameters after warmup
    Set $m_i^t = \mathbf{1}$, i.e., $m_i(\ell) = 1,\ \forall\, \ell \in [d]$
  end if
  Server sends global model x_g^{t-1} and global mask probability θ_g^{t-1} to all clients $i \in [N]$
  for client $i \in [N]$ in parallel **do**
    Initialize local model $x_i^{t,0} \leftarrow x_g^{t-1}$
    $s_i^{t,0} \leftarrow s_g^0$ **if** $t = 1$ **else** $s_i^{t,0} \leftarrow s_i^{t-1,K}$ **endif**
    for $k = 1, \ldots, K$ **do**
      Procedure I: Freeze model weights $x_i^{t,k-1}$
      Optimize over s: $\mathcal{L}_s = f_i\left(x_i^{t,k-1} \odot \mathcal{G}\left(s_i^{t,k-1}\right), \xi_i^{t,k-1}\right) - \lambda \left\| \sigma\left(s_i^{t,k-1}\right) - \theta_{g\setminus\{i\}}^{t-1} \right\|_2^2$
      Update: $s_i^{t,k} \leftarrow s_i^{t,k-1} - \eta_s \nabla_s \mathcal{L}_s$
      Procedure II. Freeze mask score vector $s_i^{t,k}$
      Optimize over x : $\mathcal{L}_x = f_i\left(x_i^{t,k-1} \odot \mathcal{G}\left(s_i^{t,k}\right), \xi_i^{t,k-1}\right)$
      Update: $x_i^{t,k} \leftarrow x_i^{t,k-1} - \eta_\ell \nabla_x \mathcal{L}_x$
    end for
    Compute $m_i^t = \mathcal{G}(s_i^{t,K})$ and upload $x_i^t \leftarrow x_i^{t,K}$, m_i^t to server
  end for
  $x_g^t = x_g^{t-1} - \eta_g \left(x_g^{t-1} - \dfrac{\sum_{i\in[N]} x_i^t \odot m_i^t}{\sum_{i\in[N]} m_i^t} \right)$
end for

In the FedAvg algorithm, the number of communication rounds necessary to achieve a certain precision is directly proportional to the heterogeneity measure [5]. Notably, this relationship holds true in convex settings; however, in non-convex scenarios, the algorithm may either not converge or may converge to a suboptimal solution. Stemming from this observation, our objective is to reduce the number of requisite communication rounds to achieve convergence, while simultaneously achieving a better solution.

4. Proposed FedPeWS Method

The core idea of the proposed `FedPeWS` method is to allow participants to learn only a personalized subnetwork (a subset of parameters) instead of the entire network (all parameters) during the initial warmup phase. Let $m_i \in \{0,1\}^d$ be a binary mask vector denoting the set of parameters that are learned by participant i, $i \in [N]$. Note that $m_i(\ell) = 1$ indicates that the ℓ^{th} element of x_i is selected for learning (value 0 indicates non-selection), $\ell \in [d]$. Thus, during the warmup phase, the objective in `FedPeWS` is to learn the parameters x along with the personalized masks m_i, i.e.,

$$\min_{x, \{m_i\}_{i\in[N]}} \frac{1}{N} \sum_{i=1}^{N} f_i(x \odot m_i), \tag{3}$$

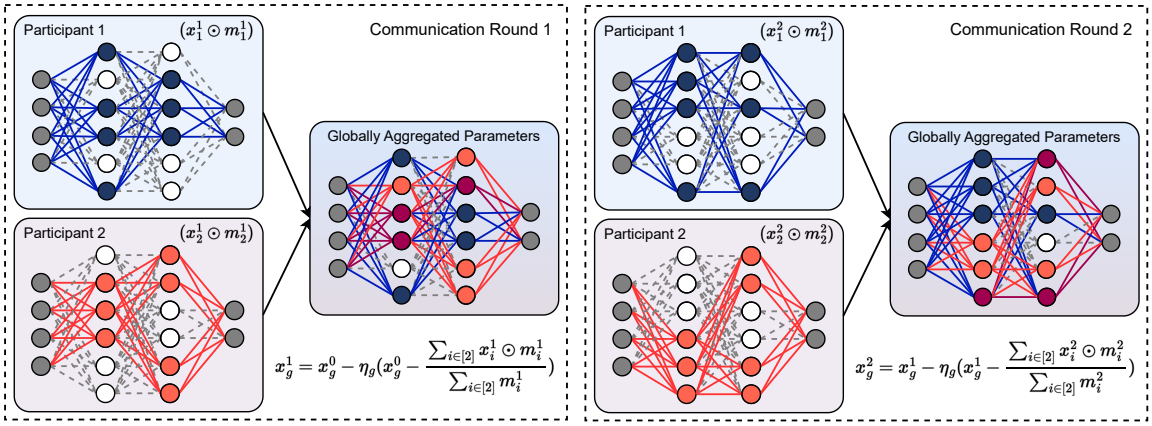


Figure 2: Illustration of the proposed FedPeWS algorithm for two participants, which aggregates partial subnetworks $(x_i^t \odot m_i^t)$ during the warmup phase to obtain a shared global model x_g^t. Here, x_i^t and m_i^t denote the local model and personalized mask of the i^{th} participant in the t^{th} round.

$\odot$ denotes element-wise multiplication. Note that $\mathcal{M}_{x\odot m_i}$ denotes the personalized subnetwork of participant i. When personalized masks are employed, the update rules can be modified as:

$$x_i^{t,k} = x_i^{t,k-1} - \eta_\ell \nabla f_i(x_i^{t,k-1} \odot m_i^t, \xi_i^{t,k-1}) \text{ and } x_g^t = x_g^{t-1} - \eta_g(x_g^{t-1} - \frac{\sum_{i\in[N]} x_i^t \odot m_i^t}{\sum_{i\in[N]} m_i^t}). \quad (4)$$

The obvious questions regarding the FedPeWS method are: (i) how to learn these personalized masks m_i? and (ii) what should be the length of the warmup period?

Identification of personalized subnetworks: It is not straightforward to directly optimize for the personalized binary (discrete) masks m_i in Equation 3. Hence, we make the following design choices. Firstly, personalized masks are learned at the neuron-level and then expanded to the parameter-level. Following IST [35], masks are specifically applied only to the hidden layer neurons, while the head and tail neurons remain unaffected. However, unlike IST, the neuron-level masks are not randomly selected in each collaboration round. Instead, we learn real-valued personalized neuron-level mask score vectors $s_i \in \mathbb{R}^h$, which in turn can be used to generate the binary masks. Here, h denotes the number of hidden neurons in the classifier $\mathcal{M}$ and $h \ll d$. A higher value of element $s_i(\ell)$, $\ell \in [h]$, indicates that the ℓ^{th} neuron is more likely to be selected by participant i. Let $\mathcal{G} : \mathbb{R}^h \to \{0,1\}^d$ be the mask generation function that generates the binary parameter-level masks m_i from neuron-level mask score vectors s_i, i.e., $m_i = \mathcal{G}(s_i)$. $\mathcal{G}$ consists of three steps. Firstly, we convert s_i into probabilities by applying a sigmoid function, i.e., $\theta_i = \sigma(s_i)$, where $\theta_i \in [0,1]^h$ is the mask probability vector and σ is the sigmoid function. Next, binary neuron masks $\tilde{m}_i$ are obtained by sampling from a Bernoulli distribution with parameter θ_i, i.e., $\tilde{m}_i(\ell) \sim Bernoulli(\theta_i(\ell)), \forall\, \ell \in [h]$. Finally, these binary neuron masks can be directly mapped to the binary parameter-level mask m_i, i.e., if a neuron is selected, all the weights associated with the selected neuron are also selected. Thus, Equation 3 can be reparameterized as:

$$\min_{x,\{s_i\}_{i\in[N]}} \frac{1}{N} \sum_{i=1}^{N} f_i(x \odot \mathcal{G}(s_i)). \quad (5)$$

The above equation can be optimized alternatively for the mask score vectors s_i and the parameters x. The participants first optimize for the mask scores while the model parameters $x_i^{t,k}$ are frozen (Procedure I), and then switch to optimizing the model parameters while freezing the mask scores (Procedure II). In the mask training step (Procedure I), the optimization objective is defined as:

$$\mathcal{L}_s = f_i\left(x_i^{t,k} \odot \mathcal{G}\left(s_i^{t,k}\right), \xi_i^{t,k-1}\right) - \lambda \left\| \sigma\left(s_i^{t,k}\right) - \theta_{g\setminus\{i\}}^t \right\|_2^2; \qquad s_i^{t,k+1} \leftarrow s_i^{t,k} - \eta_s \nabla_s \mathcal{L}_s, \quad (6)$$

where ∇_s indicates that the gradient is w.r.t. the mask score vector s, η_s is the local learning rate for updating s, θ_g^t is the global mask probability at round t, $\theta_{g\setminus\{i\}}^t$ is the global mask probability

excluding the probability mask of the current participant i, and λ is the weight assigned to the mask diversity measure (second term). It is important to note that the personalized masks may not cover all neurons in the network. Maximizing the mask diversity measure encourages personalized masks to deviate as much as possible from the global mask, which facilitates better coverage of all the neurons in the global model. The diversity measure has an upper bound due to the sigmoid function:

$$\left\| \sigma\left(s_i^{t,k}\right) - \theta_{g\setminus\{i\}}^{t} \right\|_2^2 \leq h. \tag{7}$$

Given the difficulty in calculating $\nabla_s \mathcal{L}_s$ directly due to the discrete nature of Bernoulli sampling, we employ the straight-through estimator (STE) [49] to approximate the gradients, which does not compute the gradient of the given function and passes on the incoming gradient as if the function were an identity function.

During Procedure II, the optimization function for the model weights is expressed as:

$$\mathcal{L}_x = f_i\left(x_i^{t,k} \odot \mathcal{G}\left(s_i^{t,k}\right), \xi_i^{t,k-1}\right); \qquad x_i^{t,k+1} \leftarrow x_i^{t,k} - \eta_\ell \nabla_x \mathcal{L}_x, \tag{8}$$

where ∇_x indicates that the gradient is w.r.t. weights x. The `FedPeWS` algorithm alternates between these two procedures for W rounds, where W is the number of warmup rounds. At this point, the warmup stops and the participants switch to standard training for $(T - W)$ collaboration rounds. This approach ensures that each participant effectively contributes to the FL process while also tailoring the learning to their specific data distributions. The number of warmup rounds W (or the proportion $\tau = \frac{W}{T}$) is a key hyperparameter of the `FedPeWS` algorithm, along with the weight λ assigned to the mask diversity loss. While it would be ideal to have a principled method to select these hyperparameters, we use a grid search to tune them, which is currently a limitation.

Use of fixed subnetworks: When the number of participants is small and the data distributions of the participants are known apriori, the server can partition the full model into subnetworks of the same depth and assign a fixed subnetwork to each participant, i.e., $m_i^t = m_i, \forall\, t \in [W]$. Participants transmit only the masked updates back to the server during warmup, which then aggregates these masked parameters and redistributes them in their masked form. For the sake of utility, the server can design personalized masks such that the union of these masks covers all the neurons. This variant of `FedPeWS` is referred to as `FedPeWS-Fixed` and follows the same algorithm in Algorithm 1, except for the omission of the highlighted (in green color) steps.

5. Experiments and Results

5.1. Datasets and Network Architecture

Synthetic Dataset. To effectively evaluate the performance of the proposed algorithm, we generated a custom synthetic dataset that simulates the extreme non i.i.d. scenario. This dataset encompasses four classes, each characterized by four 2D clusters determined by specific centers and covariance matrices. Note that the clusters from different classes interleave each other as shown in Figure 3. For this dataset, we utilize a neural network consisting of five fully-connected (FC) layers, each followed by ReLU activation functions, except the last layer. To enhance the dataset complexity and aid FC network learning, we transform these 2D points into 5D space using the transformation $[x, y, x^2, y^2, xy]$, based on their (x, y) coordinates. We generate two versions of this dataset, **Synthetic-32K** and **Synthetic-3.2K**, depending on the number of data points in the training set. The former has

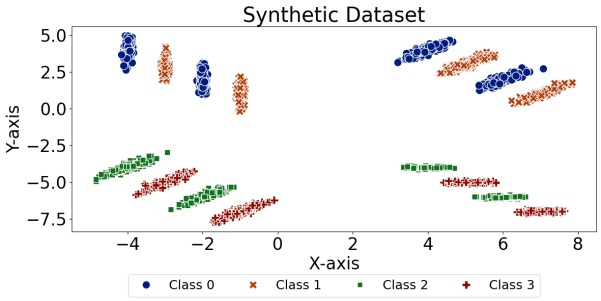


Figure 3: Samples from the custom synthetic dataset.

32000 samples, with each class containing 8000 data points, while the latter has ten times fewer data points.

CIFAR-MNIST. We integrate two distinct datasets, CIFAR-10 [50] and MNIST [51], to explore how different clients might adapt when faced with disparate data sources. CIFAR-10 comprises 32×32 pixel images categorized into 10 object classes. MNIST, typically featuring 28×28 pixel images across 10 digit classes, is upscaled to 32×32 pixel to standardize dimensions with CIFAR-10. We compile a balanced dataset by randomly selecting 400 samples from each class for the training set and 200 samples for the test set from the combined pool of 20 classes. This setup aims to simulate a FL environment where multiple clients handle significantly varied data types. For this dataset, we employ a convolutional neural network comprising four convolutional layers, each having a kernel size of 3 and padding of 1, followed by max pooling. This is succeeded by a fully connected layer. This architecture was used because of its simplicity and widespread use in the literature [20, 35].

{Path-OCT-Tissue}MNIST. We amalgamate three distinct medical datasets: PathMNIST, OCTMNIST, and TissueMNIST [52], to develop a universal medical prognosis model capable of recognizing various tasks using a single model. The datasets contain 9, 4, and 8 classes, respectively, totaling 21 classes. For this dataset, we utilized the same architecture and training details described in the CIFAR-MNIST dataset.

5.2. Experimental Setup

Dataset partitioning. For scenarios with a smaller number of collaborators ($N = 2, 3, 4$), we manually partition the training dataset to tailor the data distribution to specific participants. In the $N = 2$ scenario, we partition as follows: (i) For the Synthetic dataset, encompassing both Synthetic-32K and Synthetic-3.2K, even-numbered classes are assigned to participant 1, while odd-numbered classes are allocated to participant 2. (ii) For the CIFAR-MNIST combination, all CIFAR-10 samples are assigned to participant 1, with MNIST samples allocated to participant 2. In the $N = 3$ scenario, the {Path-OCT-Tissue}MNIST dataset is partitioned into three splits corresponding to the individual datasets, with PathMNIST assigned to participant 1, OCTMNIST to participant 2, and TissueMNIST to participant 3. For the $N = 4$ scenario, the synthetic dataset is divided so that each class is exclusively allocated to one of the four participants. For scenarios with a larger number of participants ($N \geq 10$), we employ a Dirichlet distribution to partition the training set. This approach utilizes a concentration parameter α to simulate both homogeneous and heterogeneous data distributions [28, 53–55]. We experiment with various values of α, specifically $\alpha \in \{0.1, 0.5, 1.0, 2.0, 5.0\}$, to explore the effects of dataset heterogeneity (lower α values) and homogeneity (higher α values) on the model performance. This methodological diversity allows us to comprehensively assess our approach under varying data conditions. Results for large N (> 100) are reported in the appendix.

Training details. In federated optimization, we primarily benchmark against the FedAvg algorithm [1], a standard approach in federated learning. However, our algorithm is designed to be versatile, functioning as a 'plug-and-play' solution that is compatible with various other optimizers. To demonstrate this adaptability, we also conduct experiments using FedProx [6], showcasing our method's capabilities across different optimization frameworks. Moreover, we compare these two variants to SCAFFOLD [7], FedNova [27], and MOON [56]. For our experiments, we fix the local learning rate $\eta_\ell = 0.001$ in the Synthetic-32K dataset case, and we set $\eta_\ell = 0.01$ for other experiments. Also, the mask learning rate is fixed $\eta_s = 0.1$. Furthermore, we vary the global learning rate $\eta_g \in \{0.1, 0.25, 0.5, 1.0\}$ to observe the differences in optimization behavior between the baseline and our proposed methods. Additionally, we employ two distinct batch sizes $\{32, 8\}$ for Synthetic-32K and Synthetic-3.2K, respectively. For experiments involving the CIFAR-MNIST and {Path-OCT-Tissue}MNIST datasets, we standardize the batch size to 64. We conduct our experiments on NVIDIA RTX A6000 GPUs on an internal cluster server, with each run utilizing a single GPU. The execution time for each run is capped at less than an hour, which indicates the maximum execution time rather than the average. All results are averaged over three independent runs and the average accuracy is reported on the global test dataset.

Table 2: The required number of collaboration rounds to reach target accuracy υ % and the final accuracy after T rounds. The results are averaged over 3 seeds. $\times$ indicates that the algorithm cannot reach target accuracy υ within T rounds and NA means that it reaches υ only in one random seed.

Dataset / Batch size		Synthetic-32K, 32			Synthetic-3.2K, 8
Parameters $\{\eta_g/\lambda/\tau\}$		$\{1.0/5.0/0.125\}$	$\{0.5/2.0/0.2\}$	$\{0.25/1.0/0.1875\}$	$\{0.1/2.0/0.1\}$
Target accuracy $\upsilon(\%)$		99	90	75	99
No. of rounds to reach target accuracy	FedAvg	148 ± 3.79	$199 \pm$ NA	$\times$	$371 \pm$ NA
	`FedAvg+PeWS`	$\mathbf{115 \pm 7.21}$	$\mathbf{182 \pm 6.81}$	$\mathbf{286 \pm 7.93}$	$\mathbf{301 \pm 10.59}$
Final accuracy after T collaboration rounds	FedAvg	99.94 ± 0.05	91.40 ± 7.25	67.64 ± 0.90	97.33 ± 3.89
	`FedAvg+PeWS`	$\mathbf{99.96 \pm 0.01}$	$\mathbf{99.49 \pm 0.60}$	$\mathbf{83.50 \pm 3.52}$	$\mathbf{99.66 \pm 0.19}$

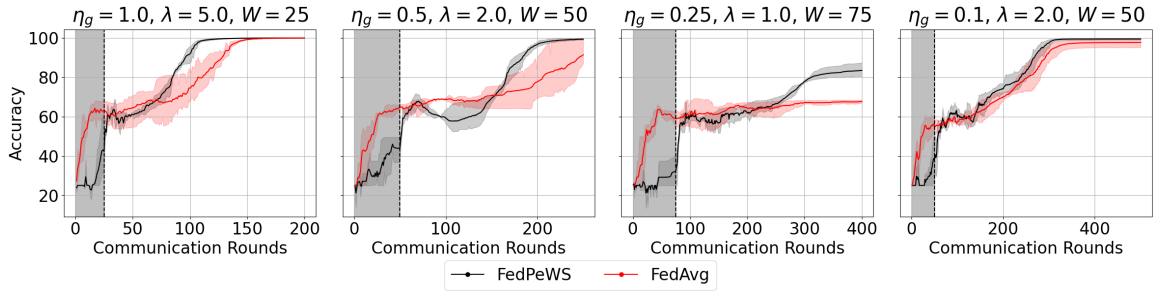


Figure 4: Results on Synthetic-{32, 3.2}K datasets with batch sizes {32, 8}, global learning rates $\eta_g \in \{1.0, 0.5, 0.25, 0.1\}$ and communication rounds $T \in \{200, 250, 400, 500\}$. See Table 2 for details. `FedPeWS` consistently converges faster and outperforms FedAvg.

5.3. Experimental Results

Our experimental analysis focuses on assessing the performance of our proposed `FedPeWS` algorithm within the FL framework. The key findings from our studies are as follows: (i) The `FedPeWS` approach demonstrates a significant reduction in the number of communication rounds required to achieve target accuracy while also enhancing the final accuracy post-convergence. (ii) The `FedPeWS` algorithm is robust across different levels of data heterogeneity. (iii) In scenarios where full knowledge of the participant data distributions is available, the server can employ `FedPeWS-Fixed` method (Figure 6).

Improved communication efficiency and accuracy. We initially report the required number of communication rounds to reach the target accuracy and the final accuracy after T communication rounds for the synthetic dataset in Table 2. The results underscore that the incorporation of a personalized warmup phase in a federated setup significantly reduces the required number of communication rounds across all tested scenarios. Notably, in specific instances, such as with the Synthetic-32K dataset and $\eta_g = 0.25$, the conventional FedAvg algorithm does not meet the target accuracy within the T communication rounds. Conversely, in scenarios where $\eta_g \in \{0.25, 0.1\}$, FedAvg only achieves the target accuracy in one of the seeds, exhibiting suboptimal performance in the other two runs. From Figure 4, it is evident that our proposed `FedPeWS` algorithm surpasses FedAvg in both communication efficiency and accuracy. We also consider a more extreme data heterogeneity scenario with $N = 4$ participants, depicted in Figure 5, where FedAvg completely fails by reaching only $58.4 \pm 2.33\%$, whereas our `FedPeWS` approach reaches $91.13 \pm 3.55\%$ accuracy by significantly outperforming the

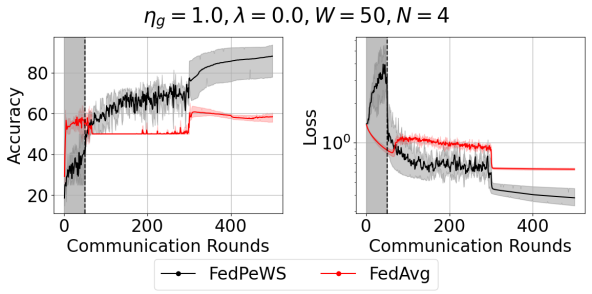
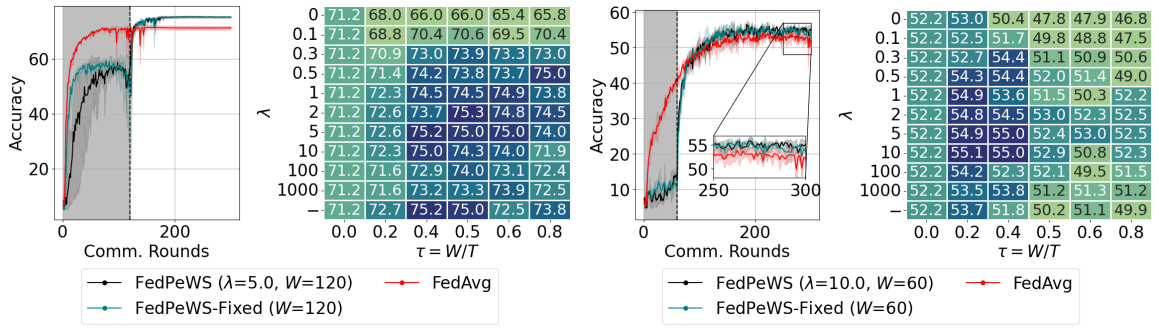


Figure 5: Visualization of validation accuracy and loss on the Synthetic-32K dataset with $N = 4$.



(a) CIFAR-MNIST dataset.

(b) {Path-OCT-Tissue}MNIST dataset.

Figure 6: Results for experiments with $T = 300$ on (a) CIFAR-MNIST and (b) {Path-OCT-Tissue}MNIST datasets. **(a)** Participant 1 uses MNIST; Participant 2 uses CIFAR-10; ablation study for λ and τ (Table 7). **(b)** $N = 3$ participants use {PathMNIST, OCTMNIST, TissueMNIST}; ablation study for λ and τ (Table 8). `FedPeWS-Fixed` results appear in the last row; $\tau = 0.0$ denotes FedAvg.

base optimizer (FedAvg) with a gain of $\mathbf{32.72}\%$. It is crucial to highlight that in this experiment, we set $\lambda = 0.0$, effectively not enforcing diversity as outlined in Equation 6. This approach focuses solely on optimizing the masks using the first loss component, which depends only on the data distributions of each participant. This shows that, in specific scenarios, we can learn the personalized masks (Procedure I) without the need to adjust the λ parameter, while still achieving a better performance than the base optimizer.

Sensitivity to λ and τ parameters. Figure 6 showcases the results of experiments on the CIFAR-MNIST dataset with $N = 2$ participants and {Path-OCT-Tissue}MNIST dataset with $N = 3$ participants. The left-side plots of Figures 6a and 6b, which depict the performance of the global model (averaged over 3 runs), demonstrate that our method consistently achieves higher accuracy. The right side figures feature heatmap plots that annotate the global model accuracy obtained varying $\lambda \in [0, 1000]$ and $\tau \in [0.0, 0.8]$ parameters. An additional row ($\lambda = -$) represents the `FedPeWS-Fixed` approach, where user(server)-defined fixed masks are employed. In this method, we simply split the full network into N partitions, with each partition assigned to a participant (for detailed instructions see Section B.2 in the appendix). The results indicate that our approach has a low sensitivity to variations in λ and τ. For more detailed insights, please refer to Tables 7 and 8 in the appendix.

Comparison to the state-of-the-art (SOTA) algorithms. We evaluate our proposed algorithm, FedPeWS, against state-of-the-art (SOTA) algorithms, with the results summarized in Table 3. As highlighted in the table, algorithms such as FedProx, SCAFFOLD, FedNova, and MOON demonstrate performance comparable to FedAvg, as illustrated in Figure 6. In contrast, our algorithm, when applied on top of FedAvg and FedProx, achieves superior performance.

Table 3: Comparison to the SOTA algorithms.

Dataset	CIFAR-MNIST	{P-O-T}MNIST
Fedavg	71.78 ± 0.66	52.83 ± 1.26
FedProx	72.27 ± 0.88	51.28 ± 1.03
SCAFFOLD	71.83 ± 0.24	53.05 ± 0.60
FedNova	71.63 ± 0.98	53.05 ± 0.83
MOON	71.84 ± 1.09	52.10 ± 0.19
`FedAvg+PeWS`	$\mathbf{75.83 \pm 0.88}$	$\mathbf{55.12 \pm 0.56}$
`FedProx+PeWS`	75.04 ± 0.85	54.67 ± 0.43

6. Conclusion

In this work, we introduced a novel concept called *personalized warmup via subnetworks* for heterogeneous FL – a strategy that enhances convergence speed and can seamlessly integrate with existing optimization techniques. Results demonstrate that the proposed `FedPeWS` approach achieves higher accuracy than the relevant baselines, especially when there is extreme statistical heterogeneity.

Acknowledgements

This material is partly based on work supported by the Office of Naval Research N00014-24-1-2168.

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] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[3] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. Matcha: Speeding up decentralized sgd via matching decomposition sampling. In *2019 Sixth Indian Control Conference (ICC)*, pages 299–300. IEEE, 2019.

[4] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with NeurIPS*, 2019.

[5] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJxNAnVtDS`.

[6] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze, editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020. URL `https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf`.

[7] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[8] Nazarii Tupitsa, Samuel Horváth, Martin Takáč, and Eduard Gorbunov. Federated learning can find friends that are advantageous. *arXiv preprint arXiv:2402.05050*, 2024.

[9] Abdurakhmon Sadiev, Ekaterina Borodich, Aleksandr Beznosikov, Darina Dvinskikh, Saveliy Chezhegov, Rachael Tappenden, Martin Takáč, and Alexander Gasnikov. Decentralized personalized federated learning: Lower bounds and optimal algorithm for all personalization modes. *EURO Journal on Computational Optimization*, 10:100041, 2022.

[10] Aleksandr Beznosikov, Vadim Sushko, Abdurakhmon Sadiev, and Alexander Gasnikov. Decentralized personalized federated min-max problems. *arXiv preprint arXiv:2106.07289*, 2021.

[11] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

[12] Samiul Alam, Luyang Liu, Ming Yan, and Mi Zhang. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in neural information processing systems*, 35: 29677–29690, 2022.

[13] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6YEQUn0QICG`.

[14] Jan Philipp Albrecht. How the gdpr will change the world. *Eur. Data Prot. L. Rev.*, 2:287, 2016.

[15] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pages 15750–15769. PMLR, 2022.

[16] Han Guo, Philip Greengard, Hongyi Wang, Andrew Gelman, Yoon Kim, and Eric Xing. Federated learning as variational inference: A scalable expectation propagation approach. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=dZrQR7OR11`.

[17] Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=GFsU8a0sGB`.

[18] Enayat Ullah, Christopher A Choquette-Choo, Peter Kairouz, and Sewoong Oh. Private federated learning with autotuned compression. In *International Conference on Machine Learning*, pages 34668–34708. PMLR, 2023.

[19] Mohammad Mahdi Rahimi, Hasnain Irshad Bhatti, Younghyun Park, Humaira Kousar, Do-Yeon Kim, and Jaekyun Moon. Evofed: Leveraging evolutionary strategies for communication-efficient federated learning. *Advances in Neural Information Processing Systems*, 36, 2023.

[20] Berivan Isik, Francesco Pase, Deniz Gunduz, Tsachy Weissman, and Zorzi Michele. Sparse random networks for communication-efficient federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=k1FHgri5y3-`.

[21] Nurbek Tastan and Karthik Nandakumar. Capride learning: Confidential and private decentralized learning based on encryption-friendly distillation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[22] Christopher A. Choquette-Choo, Natalie Dullerud, Adam Dziedzic, Yunxiang Zhang, Somesh Jha, Nicolas Papernot, and Xiao Wang. CaPC Learning: Confidential and Private Collaborative Learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=h2EbJ4_wMVq`.

[23] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1544–1551, 2019.

[24] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Advances in Neural Information Processing Systems*, 34:16104–16117, 2021.

[25] 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. *arXiv preprint arXiv:2303.16520*, 2023.

[26] Nurbek Tastan, Samar Fares, Toluwani Aremu, Samuel Horváth, and Karthik Nandakumar. Redefining contributions: Shapley-driven federated learning. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 5009–5017. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/554. URL `https://doi.org/10.24963/ijcai.2024/554`. Main Track.

[27] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[28] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

[29] Savelii Chezhegov, Sergey Skorik, Nikolas Khachaturov, Danil Shalagin, Aram Avetisyan, Aleksandr Beznosikov, Martin Takáč, Yaroslav Kholodov, and Alexander Gasnikov. Local methods with adaptivity via scaling. *arXiv preprint arXiv:2406.00846*, 2024.

[30] Elnur Gasanov, Ahmed Khaled, Samuel Horváth, and Peter Richtarik. Flix: A simple and communication-efficient alternative to local methods in federated learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11374–11421. PMLR, 28–30 Mar 2022.

[31] Filip Hanzely, Boxin Zhao, and mladen kolar. Personalized federated learning: A unified framework and universal optimization techniques. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ilHM31lXC4.

[32] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2003.03196.

[33] Ekaterina Borodich, Aleksandr Beznosikov, Abdurakhmon Sadiev, Vadim Sushko, Nikolay Savelyev, Martin Takáč, and Alexander Gasnikov. Decentralized personalized federated min-max problems. *arXiv preprint arXiv:2106.07289*, 2021.

[34] 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.

[35] Binhang Yuan, Cameron R Wolfe, Chen Dun, Yuxin Tang, Anastasios Kyrillidis, and Chris Jermaine. Distributed learning of fully connected neural networks using independent subnet training. *Proceedings of the VLDB Endowment*, 15(8), 2022.

[36] Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[37] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TNkPBBYFkXg.

[38] Bouacida Nader, Hou Jiahui, Zang Hui, and Liu Xin. Adaptive federated dropout: Improving communication efficiency and generalization for federated learning. *arXiv preprint arXiv:2011.04050*, 2020.

[39] Ang Li, Jingwei Sun, Xiao Zeng, Mi Zhang, Hai Li, and Yiran Chen. Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 42–55, 2021.

[40] Hamid Mozaffari, Virat Shejwalkar, and Amir Houmansadr. Frl: Federated rank learning. *arXiv preprint arXiv:2110.04350*, 2021.

[41] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

[42] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. Neural architecture search without training. In *International conference on machine learning*, pages 7588–7598. PMLR, 2021.

[43] Shaoxiong Ji, Wenqi Jiang, Anwar Walid, and Xue Li. Dynamic sampling and selective masking for communication-efficient federated learning. *IEEE Intelligent Systems*, 37(2):27–34, 2021.

[44] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020.

[45] Tiansheng Huang, Shiwei Liu, Li Shen, Fengxiang He, Weiwei Lin, and Dacheng Tao. Achieving personalized federated learning with sparse local models. *arXiv preprint arXiv:2201.11380*, 2022.

[46] Sara Babakniya, Souvik Kundu, Saurav Prakash, Yue Niu, and Salman Avestimehr. Federated sparse training: Lottery aware model compression for resource constrained edge. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022. URL https://openreview.net/forum?id=XEQSP1zL2gx.

[47] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[48] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[49] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

[50] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Toronto, ON, Canada*, 2009.

[51] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[52] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

[53] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

[54] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1484–1490. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/205. URL https://doi.org/10.24963/ijcai.2021/205. Main Track.

[55] 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.

[56] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

Contents

A. Scalability of the Mask Mechanism

Our approach is designed as a plug-and-play extension to existing federated optimization methods with minimal overhead.

To estimate the computational overhead, let us assume that a baseline method such as FedAvg incurs a training cost of 1 unit per communication round. In our method, a warm-up phase occupies a fraction τ of the total communication rounds (e.g., $\tau \in \{0.1, 0.5\}$). Let ω be the computational complexity of each warmup round. Consequently, the total cost of the overall training will increase only when ω is greater than 1. During the warmup rounds, additional costs from mask learning are incurred. It must be noted that we learn only neuron-level masks, and the number of hidden neurons is much lower than the number of parameters. On the other hand, these additional mask learning costs are offset by the fact that only sparse subnetworks are learned during the warmup rounds. Empirically, the observed sparsification level lies in the range $[0.1, 0.5]$. Under the worst-case scenario of sparsification at 0.5, these two factors (computational cost increase due to mask learning and computational cost decrease due to sparsification) cancel each other out and $\omega \approx 1$. Therefore, the computational cost of `FedPeWS` is approximately the same as FedAvg when sparsification is 0.5. For more aggressive sparsification (< 0.5), `FedPeWS` becomes more computationally efficient than FedAvg.

B. Additional Experimental Details

B.1. Network Architecture Details

The network for the synthetic dataset (detailed in Section 5.2) consists of five fully connected (FC) layers, each followed by ReLU activation functions, except for the last layer. We provide the details of this architecture in Table 4.

Table 4: Architecture for synthetic dataset models used in the experiments.

Layer	Input	FC1	FC2	FC3	FC4	FC5
Dimensions	[5]	[5, 32]	[32, 64]	[64, 128]	[128, 32]	[32, 4]

The network for the CIFAR-MNIST and {Path-OCT-Tissue}MNIST datasets includes three convolutional layers followed by max pooling, and a fully connected layer. The details of this architecture is provided in Table 5.

Table 5: Architecture for CIFAR-MNIST dataset models. Every convolutional layer is followed by a max pooling layer with kernel size 2 and stride 2.

Layer	Input	Conv1	Conv2	Conv3	Flatten	FC
Dimensions	[3, 32, 32]	[3, 32, 3, 3]	[32, 64, 3, 3]	[64, 128, 3, 3]	[2048]	[2048, 20]

B.2. Fixed Mask Generation

Figure 7 illustrates how we design masks for `FedPeWS-Fixed` experiments in scenarios with $N = 2$ participants. For cases involving $N = 4$ participants, the full network $\mathcal{M}_x$ (classifier $\mathcal{M}$ parameterized with x) is divided into four subnetworks, vertically, with each subnetwork corresponding to one of the participants. As such, we vertically partition the hidden neurons in the network into N groups (subnetworks) and design the mask to assign each group to one participant, ensuring no overlap. This design choice is based on the assumption that classes held by each participant are highly heterogeneous, thus preventing any intersection in the masks. This setting is specifically tailored for the `FedPeWS-Fixed` method and doesn't necessitate performing optimization over the masks m_i; they are kept fixed.

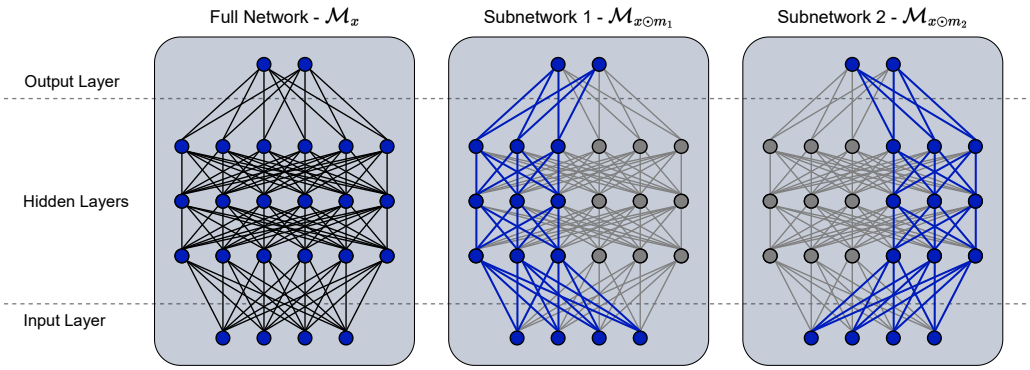


Figure 7: Illustration of manual mask setting in the `FedPeWS-Fixed` method. The left figure illustrates the complete network with all neurons active and full connections. The middle figure represents subnetwork 1, utilizing only the left portion of the full network, where m_1 corresponds to this left side. Conversely, the right figure indicates the part of the network used for subnetwork 2. This setting is employed in all experiments involving $N = 2$ participants.

C. Experimental Results

C.1. Varying Degrees of Heterogeneity

Figure 8 demonstrates that our `FedPeWS` approach consistently outperforms FedAvg, with gains directly related to the degree of data heterogeneity. The figure clearly shows that the advantage of using our method is more pronounced under conditions of high data heterogeneity. As heterogeneity levels decrease, our method becomes comparable to FedAvg.

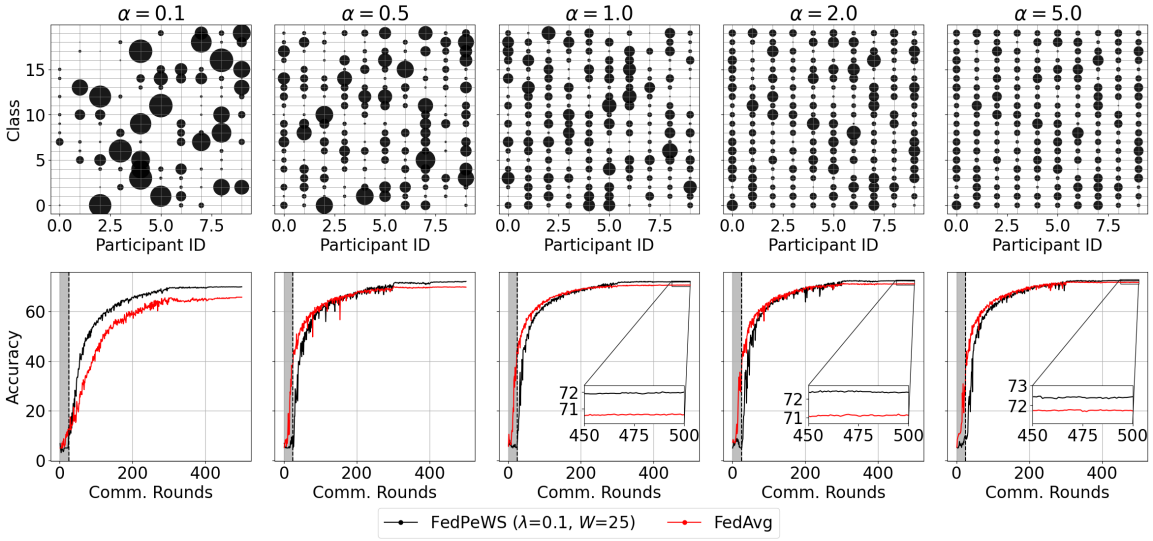


Figure 8: Top: illustration of number of samples per class allocated to each client, that is indicated by dot sizes, for different concentration α values. Bottom: visualization of the experiments on CIFAR-MNIST dataset with $N = 10$ participants with different levels of heterogeneity.

C.2. Comparison to FedProx

We also present results using the FedProx optimizer on Synthetic-32K and Synthetic-3.2K datasets in Figure 9 and Table 6, employing global learning rates $\eta_g = \{1.0, 0.5, 0.25, 0.1\}$. Note that we adapt Algorithm 1 to incorporate the FedProx algorithm as the base optimizer, instead of FedAvg.

Table 6: The required number of collaboration rounds to reach target accuracy υ % using FedProx algorithm and the final accuracy after T rounds. The results are averaged over 3 seeds. × indicates that the algorithm cannot reach target accuracy υ within T rounds.

Dataset / Batch size		Synthetic-32K, 32			Synthetic-3.2K, 8
Parameters $\{\eta_g/\lambda/\tau\}$		$\{1.0/0.1/0.125\}$	$\{0.5/0.1/0.2\}$	$\{0.25/1.0/0.1875\}$	$\{0.1/1.0/0.1\}$
Target accuracy $\upsilon(\%)$		99	90	75	99
No. of rounds to reach target accuracy	FedProx	138 ± 13.22	×	×	362 ± 20.00
	`FedProx+PeWS`	$\mathbf{115 \pm 5.29}$	$\mathbf{211 \pm 16.52}$	$\mathbf{314 \pm 27.83}$	$\mathbf{344 \pm 27.30}$
Final accuracy after T collaboration rounds	FedProx	99.95 ± 0.02	82.43 ± 7.98	69.26 ± 6.03	$\mathbf{99.92 \pm 0.06}$
	`FedProx+PeWS`	$\mathbf{99.96 \pm 0.01}$	$\mathbf{98.40 \pm 1.84}$	$\mathbf{90.40 \pm 3.91}$	$\mathbf{99.92 \pm 0.07}$

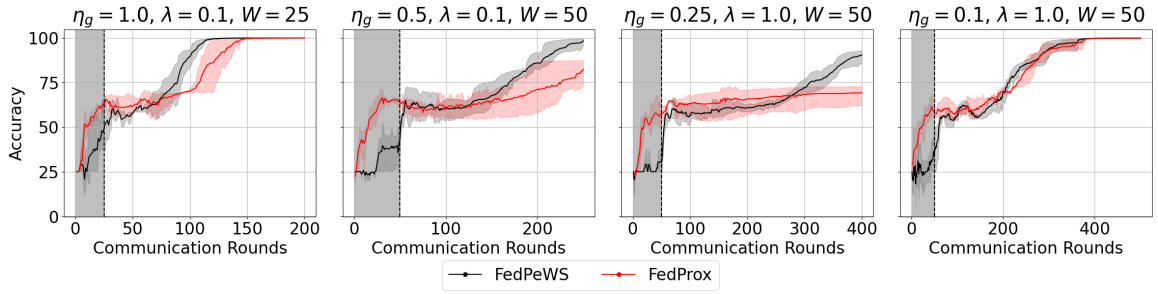


Figure 9: Comparison of our proposed method and FedProx [6] on Synthetic-{32, 3.2}K datasets. Refer to Table 6 for the corresponding numbers.

We selected the best performing proximal term scaler 0.01 after tuning and evaluating different values from a set of potential values $\{0.001, 0.01, 0.1, 0.5\}$, based on the findings in [6]. The results demonstrate that `FedPeWS` outperforms FedProx in terms of both communication efficiency and final accuracy across the tested scenarios, except the last scenario (Synthetic-3.2K dataset with batch size 8 and $\eta_g = 0.1$), where the performance of `FedPeWS` is comparable to that of FedProx.

C.3. Wall-clock time vs. Accuracy

Figure 10 illustrates the wall-clock time versus accuracy results, which correspond to Figure 4 in the main paper. From this comparison, `FedPeWS` demonstrates a slightly improved performance over FedAvg in terms of wall-clock time in two of the four scenarios. However, it underperforms slightly in the remaining two scenarios, with only a marginal increase in time. This variance is attributed to the alternation between training masks and weights during the warmup phase, impacting the time efficiency.

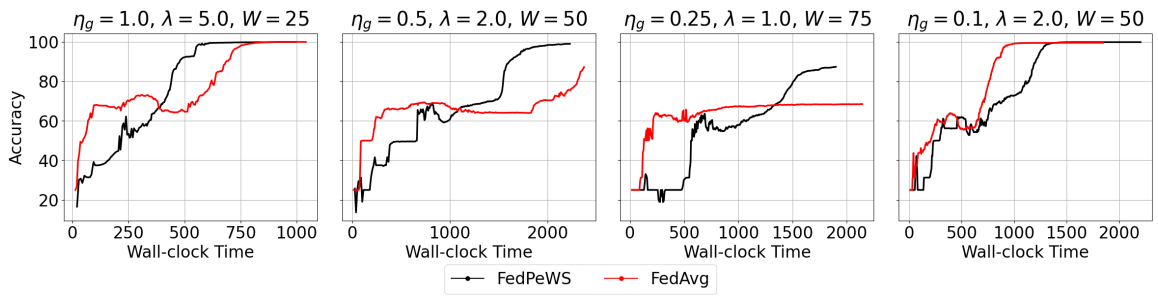


Figure 10: Wall-clock time vs. accuracy plot corresponding to Figure 4 of the main paper. For better clarity, we selected a single run for each experiment at random.

C.4. `FedPeWS-Fixed`. Mask Length Study

In this section, we explore the impact of mask length on the performance of the `FedPeWS-Fixed` method with parameter $W = 120$ ($\tau = 0.4$) using the CIFAR-MNIST dataset. We examine two scenarios for splitting the network into two subnetworks:

1. $|m_1| < |m_2|$: 75% of the mask is assigned to Participant 2, 25% to Participant 1.
2. $|m_1| = |m_2|$: equal sized masks are assigned to each participant.

Figure 11 displays the validation accuracy over $T = 300$ communication rounds for both scenarios. The leftmost plot shows the accuracy of the global model, while the middle and rightmost plots show the accuracy for each of the participants. Both mask length scenarios converge to comparable accuracy levels, with a marginal difference of 0.5% higher accuracy in the scenario where $|m_1| < |m_2|$. This is likely due to the larger mask size, which aids in learning the more complex CIFAR-10 dataset held by Participant 2.

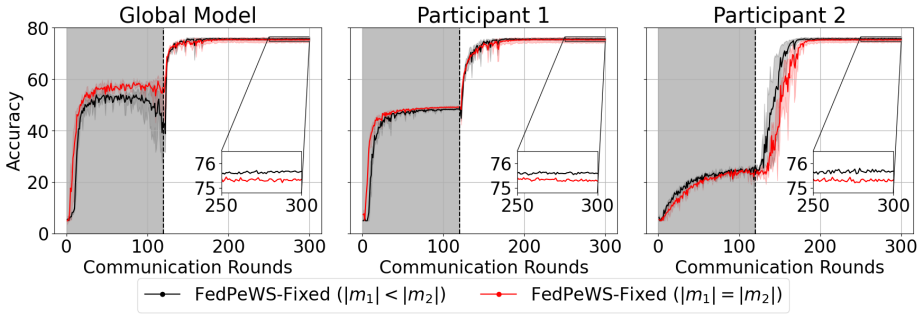


Figure 11: Mask length study using `FedPeWS-Fixed` method on CIFAR-MNIST dataset.

C.5. Sensitivity Analysis

In this section, we detail the sensitivity analysis of the λ and τ parameters conducted on the CIFAR-MNIST dataset (Table 7) and the {Path-OCT-Tissue}MNIST dataset (Table 8). This analysis particularly includes the standard deviation of the accuracy achieved by the tested algorithms after T communication rounds and over three independent evaluations. Results with the best performance are highlighted in green.

Table 7: Ablation study of the parameters λ and τ on the CIFAR-MNIST dataset with $N = 2$ participants over three independent runs. The first column ($\tau = 0.0$) corresponds to the FedAvg algorithm. The last row ($-$) presents results for the `FedPeWS-Fixed` algorithm.

	Proportion of warmup rounds $\tau = W/T$					
$\lambda(\downarrow), \tau(\rightarrow)$	$\tau = 0.0$ (FedAvg)	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.8$
0.0		68.01 ± 0.88	66.01 ± 0.48	65.96 ± 1.03	65.40 ± 1.95	65.77 ± 0.41
0.1		68.77 ± 1.09	70.39 ± 1.00	70.61 ± 1.17	69.48 ± 0.37	70.36 ± 2.06
0.3		70.86 ± 0.23	73.00 ± 0.65	73.91 ± 0.71	73.26 ± 0.46	73.02 ± 1.05
0.5		71.43 ± 0.56	74.17 ± 0.91	73.84 ± 0.12	73.66 ± 0.88	75.05 ± 0.45
1.0		72.26 ± 0.54	74.46 ± 0.44	74.54 ± 0.76	74.91 ± 0.42	73.81 ± 0.87
2.0	71.23 ± 0.71	72.61 ± 0.79	73.68 ± 0.17	$\mathbf{75.35 \pm 0.50}$	74.76 ± 0.54	74.46 ± 0.56
5.0		72.60 ± 0.45	$\mathbf{75.22 \pm 0.33}$	$\mathbf{75.00 \pm 0.74}$	75.01 ± 0.71	73.96 ± 1.60
10.0		72.29 ± 0.48	$\mathbf{74.97 \pm 0.65}$	74.31 ± 0.95	74.03 ± 0.30	71.91 ± 2.69
100.0		71.64 ± 0.47	72.92 ± 0.39	73.96 ± 0.65	73.13 ± 0.73	72.43 ± 3.55
1000.0		71.58 ± 0.53	73.18 ± 0.73	73.32 ± 1.70	73.87 ± 1.16	72.52 ± 1.92
$-$		72.72 ± 0.44	$\mathbf{75.22 \pm 0.19}$	$\mathbf{75.05 \pm 0.42}$	72.51 ± 3.89	73.77 ± 0.20

Table 8: Ablation study of the parameters λ and τ on the combination of {Path-OCT-Tissue}MNIST datasets with $N = 3$ participants over three independent runs. The first column ($\tau = 0.0$) corresponds to the FedAvg algorithm. The last row ($-$) presents results for the `FedPeWS-Fixed` algorithm.

	Proportion of warmup rounds $\tau = W/T$					
$\lambda(\downarrow), \tau(\rightarrow)$	$\tau = 0.0$ (FedAvg)	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.8$
0.0	52.25 ± 0.57	53.04 ± 2.08	50.35 ± 0.61	47.83 ± 1.04	47.89 ± 1.17	46.80 ± 1.76
0.1		52.52 ± 1.79	51.72 ± 0.63	49.83 ± 1.83	48.83 ± 2.51	47.50 ± 1.07
0.3		52.74 ± 1.75	54.36 ± 0.80	51.07 ± 0.25	50.94 ± 2.22	50.62 ± 1.91
0.5		54.30 ± 2.08	54.42 ± 1.43	52.02 ± 2.53	51.43 ± 1.60	48.97 ± 0.80
1.0		$\mathbf{54.89 \pm 0.72}$	53.59 ± 1.40	51.49 ± 1.19	50.29 ± 2.70	52.21 ± 1.02
2.0		$\mathbf{54.75 \pm 1.12}$	$\mathbf{54.45 \pm 1.78}$	53.02 ± 0.45	52.26 ± 1.81	52.49 ± 0.49
5.0		$\mathbf{54.91 \pm 0.91}$	$\mathbf{54.99 \pm 0.90}$	52.41 ± 0.45	52.95 ± 1.49	52.53 ± 1.35
10.0		$\mathbf{55.12 \pm 1.16}$	$\mathbf{55.03 \pm 1.39}$	52.85 ± 0.95	50.82 ± 3.03	52.27 ± 1.11
100.0		54.22 ± 1.74	52.31 ± 2.55	52.10 ± 0.87	49.52 ± 4.34	51.46 ± 2.92
1000.0		53.45 ± 1.65	53.82 ± 2.16	51.16 ± 1.92	51.30 ± 2.02	51.19 ± 1.05
$-$		53.69 ± 0.77	51.78 ± 0.44	50.24 ± 1.88	51.12 ± 0.72	49.87 ± 1.23

For the CIFAR-MNIST dataset, the preferred values of λ that yield optimal outcomes range within $\{2.0, 5.0, 10.0\}$, and for τ, the values are $\{0.4, 0.5\}$. A similar pattern is observed in the {Path-OCT-Tissue}MNIST dataset experiment, with a small difference, in which it shows a preference for fewer warmup rounds ($\tau \in \{0.2, 0.4\}$) and demonstrates optimal performance with the same set of λ values. This consistency across different datasets indicates robustness in the parameter settings for achieving high accuracy.

C.6. Larger number of participants

C.6.1. $\mathbf{N = \{5, 10, 20\}}$ participants.

Figure 12 demonstrates the performance of our proposed algorithm on the CIFAR-10 and CIFAR-100 datasets [50] using a non-overlapping class splitting strategy, where an equal number of classes are sampled without replacement, totaling to the number of participants multiplied by the number of classes assigned to each participant. This represents a heterogeneous setting. For this comparison, we use FedProx with a scalar multiplier of 0.01 as the baseline algorithm. The FedPeWS algorithm combines FedProx and our proposed method, referred to as FedProx+PeWS. As evident from the plots, our algorithm consistently outperforms the baseline, delivering a significant performance boost in heterogeneous scenarios and showcasing its effectiveness across various settings.

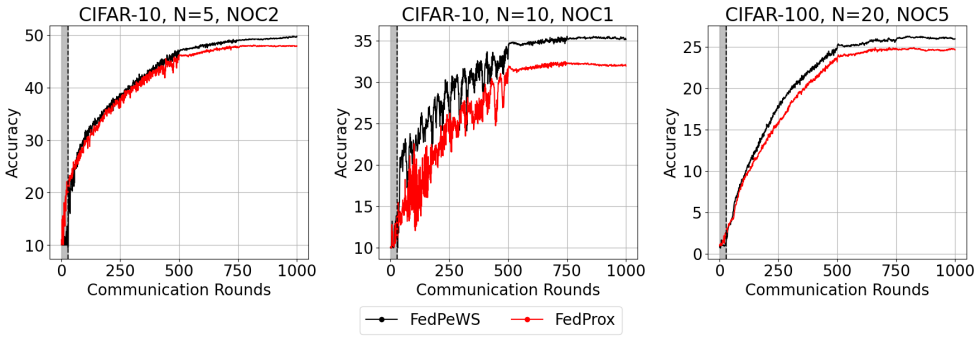


Figure 12: Visualization of global model performance with $N = 5, 10, 20$ participants on the CIFAR-10 and CIFAR-100 datasets. ***Note:*** NOC refers to non-overlapping classes partition; e.g. for the CIFAR-100 experiment, we randomly select 5 classes out of 100 for each of the 20 participants without replacement.

C.6.2. N = 200 participants.

Although our primary focus is on the cross-silo setting, we extend our study to include a large-scale scenario involving 200 participants on the CIFAR-MNIST dataset. We adopt a Dirichlet partition strategy with concentration parameter $\alpha = 0.5$ and implement this scenario with a partial participation rate of 0.1. The outcomes of this experiment, as depicted in Figure 13, indicate a superior performance compared to the conventional FedAvg algorithm, thereby further substantiating the validity and effectiveness of our proposed method. The parameters set for `FedPeWS` are: $W = 25$ and $\lambda = 0.5$.

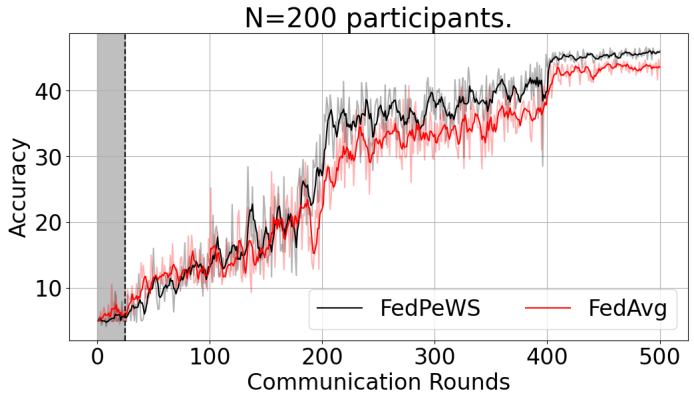


Figure 13: Visualization of global model performance with $N = 200$ participants with a partial participation rate of 0.1. Smoothing is applied as a running average with a window size of 5. A learning rate scheduler is implemented at rounds 200 and 400 with a learning rate decay factor of 0.1.

D. Neuron Activations Study

In this section, we examine the extent to which neurons in each layer are activated. Our study uses the Synthetic-32K dataset and the `FedPeWS-Fixed` method (with parameter $W = 50$). The vertical dashed line ($W = 50$) indicates the point at which participants switch to using full masks.

Figure 14 displays the neuron activations, measured as the sum of activations over a batch of samples randomly selected from each participant's dataset, over $T = 250$ communication rounds. The top row shows the outcomes for Participant 1, and the bottom row shows the activations for Participant 2. Each column corresponds to different fully connected layers (FC1 to FC4) in the network. Observations are as follows:

1. Before switching ($t \leq W$): for Participant i, subnetwork i shows higher activation patterns in all given FC layers, $i \in [1, 2]$, while the other subnetwork exhibits a minimal activation.
2. After switching to full mask ($t > W$): (i) there is a noticeable increase in activations for both participants upon switching to using full masks, (ii) Participant 1 with its originally initialized subnetwork 1, shows a substantial increase in activations compared to subnetwork 2 across all layers. The same pattern is observed for Participant 2 with subnetwork 2.

These findings suggest that the personalized warmup strategy helps the network learn which paths to follow when specific data points are fed into the network. This supports the superiority of our method and corroborates the claims made in the main paper. On the other hand, for the FedAvg algorithm (see Figure 15), we observe the absence of a similar activation pattern. Data points from different participants activate overlapping regions of the model, which results in slower convergence.

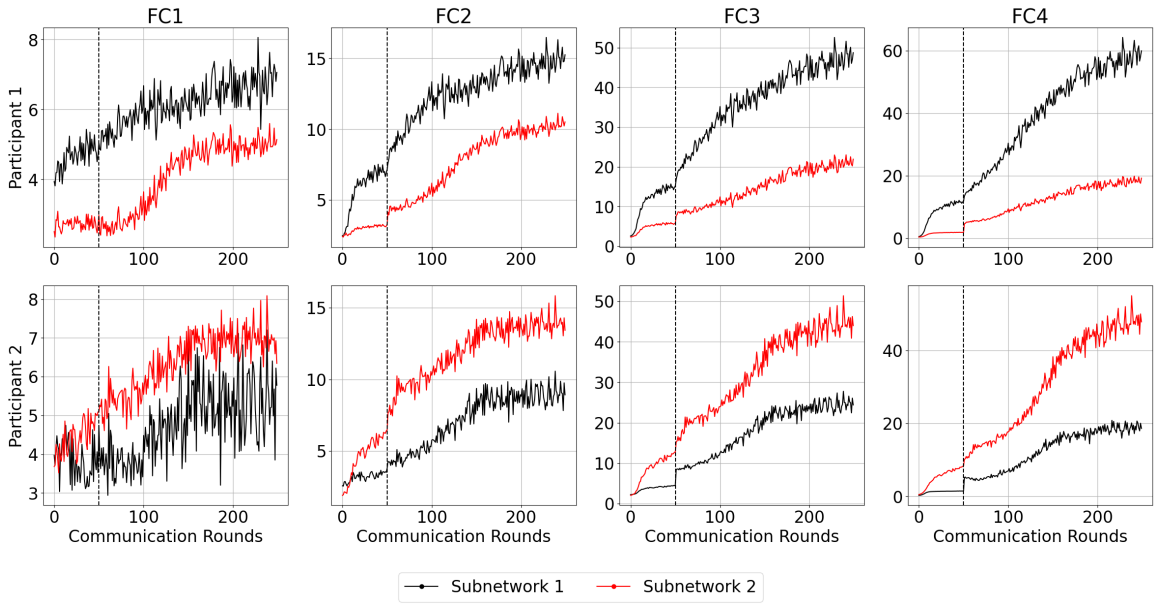


Figure 14: Neuron activation study on the Synthetic-32K dataset with a global learning rate $\eta_g = 1.0$. The experiment uses the `FedPeWS-Fixed` method with parameter $W = 50$, indicated by the vertical dashed line, marking the switch to full masks by each participant.

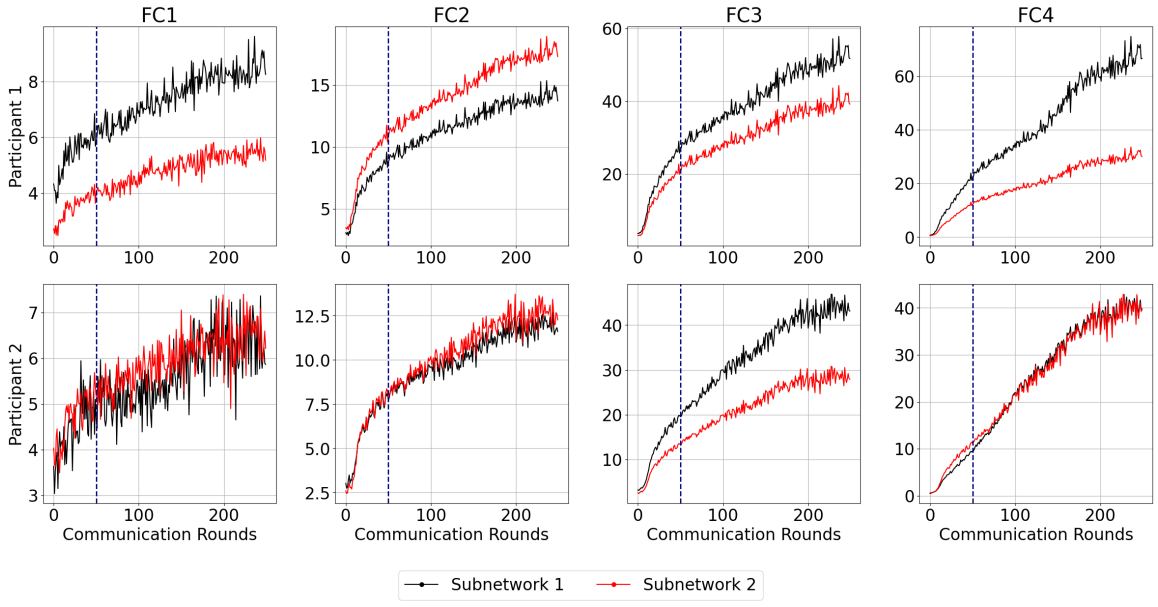


Figure 15: Neuron activation study on the Synthetic-32K dataset with a global learning rate $\eta_g = 1.0$ using FedAvg algorithm.