FedSelect: Personalized Federated Learning with Customized Selection of Parameters for Fine-Tuning

Rishub Tamirisa[†], Chulin Xie[†,‡], Wenxuan Bao[†,‡], Andy Zhou[†], Ron Arel[†], Aviv Shamsian[§]
[†]Lapis Labs [‡]University of Illinois Urbana-Champaign [§]Bar-Ilan University
{rishubt2,chulinx2,wbao4,andyz3,ronarel2}@illinois.edu aviv.shamsian@live.biu.ac.il

Abstract

Standard federated learning approaches suffer when client data distributions have sufficient heterogeneity. Recent methods addressed the client data heterogeneity issue via personalized federated learning (PFL) - a class of FL algorithms aiming to personalize learned global knowledge to better suit the clients' local data distributions. Existing PFL methods usually decouple global updates in deep neural networks by performing personalization on particular layers (i.e. classifier heads) and global aggregation for the rest of the network. However, preselecting network layers for personalization may result in suboptimal storage of global knowledge. In this work, we propose FEDSELECT*, a novel PFL algorithm inspired by the iterative subnetwork discovery procedure used for the Lottery Ticket Hypothesis.* FEDSELECT *incrementally expands subnetworks to personalize client parameters, concurrently conducting global aggregations on the remaining parameters. This approach enables the personalization of both client parameters and subnetwork structure during the training process. Finally, we show that* FEDSELECT *outperforms recent state-of-the-art PFL algorithms under challenging client data heterogeneity settings and demonstrates robustness to various real-world distributional shifts. Our code is available at https://github.com/lapisrocks/fedselect.*

1. Introduction

Federated Learning (FL) enables distributed clients or devices to jointly learn a shared global model while keeping the training data local [31], which is particularly beneficial for privacy-critical applications [37]. Despite its potential, the efficacy of FL is challenged by the inherent heterogeneity of data across different clients. As the local updates of clients can be remarkably diverse given their heterogeneous local data, the aggregated global model can diverge under the standard FL paradigms [23].

To address this issue, personalized federated learning (PFL) emerges as a promising approach. PFL allows individual clients to maintain their unique, personalized models, with parameters tailored to their local data distributions, while knowledge sharing across different clients. One representative type of PFL algorithm is "parameter decoupling". It decomposes the FL model into two distinct components: a global subnetwork shared among all clients (e.g., the feature extractor), and a personalized subnetwork adapted to each client's local distribution (e.g., the prediction head). By personalizing a subset of parameters, parameter decoupling strikes a balance between knowledge sharing among clients and personalization to each client.

A natural follow-up question would be *how to determine which parameters to personalize*. Previous works typically selected specific layers for personalization, e.g., the prediction head [5, 7], and the feature extractor [27, 33]. However, recent studies [11, 12, 35] suggest that even within the same layer, the importance of parameters for prediction can vary significantly. The coarse-grained layer-wise selection of personalized subnetwork may not fully balance knowledge sharing among clients and personalization to each client. Furthermore, current methods typically pre-defining the architecture of the personalized subnetwork, i.e., which layers to personalize, and the architecture of the personalized subnetwork is shared for all clients. These designs limit the performance of parameter decoupling, as the optimal personalized subnetwork often varies depending on each client's specific local data distribution.

In this paper, we delve into these challenges, exploring a novel strategy named FEDSELECT for parameter selection and subnetwork personalization in PFL to unlock the full potential of parameter decoupling. Our method in enlightened by the Lottery Ticket Hypothesis (LTH): deep neural networks contain subnetworks (i.e., winning tickets) that can reach comparable accuracy as the original network. We believe that FL model also contain a subnetwork that is crucial for personalization to a specific client's local distribution, and personalizing that subnetwork can achieve optimal balance between global knowledge sharing and local personalization. Specifically, we believe that parameters

that change the most over the course of a local client update should be personalized, while parameters changing the least should be globally aggregated. This is achieved by comparing the element-wise magnitude difference between the state of the client model before and after local training.

Notably, our method is different from the original LTH in twofold: in terms of purpose, while LTH seeks a sparse network for efficiency, our goal is to enhance personalized FL performance by finding the optimal subnetwork of individual clients for personalization, based on the characteristics of local data distribution. Methodologically, LTH prunes less important parameters (i.e., non-winning tickets) to zero value, whereas we assign them the global aggregated value for storing global information, aligning with the collaborative nature of FL. We summarize our contributions below:

- We introduce a hypothesis for selecting parameters during training time, aimed at enhancing client personalization performance in FL.
- We propose FEDSELECT, a novel personalized FL algorithm based on our hypothesis that automatically identifies the customized subnetwork for each client's personalization, guided by the updating magnitude of each parameter.
- We evaluate FEDSELECT across a range of benchmark datasets: CIFAR10, CIFAR10-C, OfficeHome, and Mini-ImageNet. Our method outperforms state-of-the-art personalized FL baselines in various FL scenarios, encompassing feature and label shifts, as well as different local data sizes. Moreover, our visualizations verify that the learned personalized subnetworks successfully capture the distributional information among clients.

2. Related Work

2.1. Federated Learning

There has been extensive work in federated learning on enhancing the training of a global model from various clients with non-independent and identically distributed (non-IID) data [4, 22]. The original approach, FedAvg [31], aims to develop a single global model computed through the aggregation of local updates from each client without requiring raw client data storage at a central server. However, challenges arise with non-IID data, leading to innovations in optimizing local learning algorithms through objective regularization [1], local bias correction [24], and data handling techniques such as class-balanced resampling [16] or loss reweighting [43]. Different from these works, we adapt each client to its local data.

2.2. Personalized Federated Learning

Personalized federated learning aims to address the issue of data heterogeneity by adapting clients to their local data distribution [40]. Common approaches include multitask learning [3, 38], clustering [9, 13, 29], transfer learning [46, 47], meta learning [6, 10, 17], hypernetworks [36], and gaussian processes [2]. We focus on partial model personalization, which seeks to improve the performance of client models by altering a subset of their structure or weights to better suit their local tasks. It also addresses the issue of "catastrophic forgetting" [30], an issue in personalized FL where global information is lost when fine-tuning a client model on its local distribution from a global initialization [18, 34]. It does this by forcefully preserving a subset of parameters, u, to serve as a fixed global representation for all clients. However, existing methods [7, 34, 44] introduced for partial model personalization require hand-selected partitioning of these shared and personalized parameters and choose u as only the input or output layers.

LotteryFL [21] learns a shared global model via FedAvg [31] and personalizes client models by pruning the global model via the vanilla LTH. Importantly, parameters are pruned to zero according to their magnitude after an iteration of batched stochastic gradient updates. However, due to a low final pruning percentage in LotteryFL, the lottery tickets found for each client share many of the same parameters, and lack sufficient personalization [32].

3. Background

Preliminaries We start by introducing the standard FL setting. Let the set of clients be $C = \{c_1, \ldots, c_N\}$, where the total number of clients $N = |C|$. For the k-th client $c_k \in C$, the corresponding local dataset is defined as $\mathcal{D}_i = \{x_i^k, y_i^k\}_{i=1}^{N_k}$, where N_k is the number of local data points for client c_k. Let θ_g be the FL global model, and the local loss function (e.g. cross-entropy loss) for client k as $f_k(\theta_g, x)$, we can define the canonical FL objective:

$$\min_{\theta_g} \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{N_k} f_k(\theta_g, x_i^k) \tag{1}$$

FEDAVG [31] optimizes the objective in (1) by locally training each of the client models θ_k^t for L number of local epochs at each communication round t. The updated local client model $\theta_k^{t^+}$ is aggregated into the global model θ_g^{t+1}, resulting in the global model update for round t in FEDAVG being $\theta_g^{t+1} \leftarrow \frac{1}{N} \sum_k^N \theta_k^{t^+}$. At the next round, the aggregated global model θ_g^{t+1} is then redistributed to all clients, resulting in $\theta_k^{t+1} \leftarrow \theta_g^{t+1}, \forall k$.

The goal of personalized federated learning, on the other hand, is to find personalized models θ_k for each client k, either adapted from θ_g or discovered independently, resulting in a modified objective:

$$\min_{\theta_1, \theta_2, \ldots, \theta_N} \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{N_k} f_k(\theta_k, x_i^k) \tag{2}$$

Partial model personalization refers to the procedure in which model parameters are partitioned into shared and personalized parameters, denoted u and v, for averaging and personalization. We consequently define $\theta_k = (u_k, v_k)$, where u denotes a set of shared global parameters (e.g., aggregated via FEDAVG), and v_k the personalized client parameters. Substituting the tuple (u_k, v_k) for each client model θ_k yields the partial model personalization objective as in [34].

4. FedSelect

In this section we discuss the motivation for our algorithm, followed by a description of the top-level procedure (Algorithm 1). We then provide detailed overviews of 2 sub-procedures within FEDSELECT (Algorithms 2, 3). We present an analogy between FEDSELECT and the algorithm used by [21] for finding winning lottery ticket networks, and note important distinctions in our application to FL.

4.1. Motivation

Prior works in FL selectively personalize specific layers of the model to better learn the heterogeneous distributions of local clients [7, 20, 26, 27, 34], which are referred to as performing "parameter-decoupling". For example, they rely on the conventional wisdom that the final linear layer of deep neural networks (DNNs) for classification contains a more enriched semantic representation with respect to the output prediction neurons [5, 45], which could be more suitable for personalization.

In this work, we propose a novel hypothesis describing that only *parameters* that change the most during training are necessary as personalized parameters for client models during FL; the rest can be aggregated as global parameters. Analogous to the sparse winning ticket networks found via the LTH, FEDSELECT aims to elicit an optimal subnetwork for *fine-grained personalization* of individual client model. Our primary intuition is that drastic distributional changes in the client personalization task may be better accommodated by preserving the accumulated global knowledge and personalized knowledge in a *parameter-wise* granularity, rather than *layer-wise*. Following this we state our hypothesis:

FL Gradient-based Lottery Ticket Hypothesis. *When training a client model on its local distribution during federated learning, parameters exhibiting minimal variation are suitable for encoding shared knowledge, while parameters demonstrating significant change are optimal for fine-tuning on the local distribution and encoding personalized knowledge.*

Now we introduce some basic notations regarding neural subnetwork and vanilla LTH. Given a neural network with parameters θ, we define a subnetwork via a binary mask m. For convenience, we use the Hadamard operator $\odot$ to be an indexing operator for m, rather than an elementwise multiply operator. For example, $\theta \odot m$ assumes θ and m have the same dimensions and returns a reference to the set of parameters in θ where m is not equal to zero. We also define the operator $\neg$ to invert the binary masks. To identify winning tickets via the vanilla LTH, the iterative magnitude search (IMS) procedure is proposed [11], which finds a mask m for network parameters θ such that $\theta \odot m$ can be trained in isolation to match the performance of θ on a given dataset. For clarity, we restate the IMS procedure as follows. Given a pruning rate p, an initial model θ is trained for j iterations, yielding θ^+. Next, the smallest $p\%$ of parameters in θ^+ are identified via a binary mask m. The original model θ is then pruned, given by $\theta \odot \neg m \leftarrow 0$, and the process repeats until the desired sparsity of m is achieved.

While we draw inspiration from the winning ticket IMS, there are three distinct and notable differences in our approach that set it apart, which will be covered in greater detail in Sections 4.2 and 4.4. First, rather than iteratively finding smaller subnetworks, our approach iteratively *grows* a subnetwork within each client. Second, we do not perform any parameter pruning; instead, we use m_k for each client θ_k to obtain the partition of global and personalized parameters as $(u_k, v_k) = (\theta_k \odot \neg m_k, \theta_k \odot m_k)$. Third, the mask pruning update to m_k is computed based on the element-wise magnitude of parameter update ∇v_k from local client training rather than the magnitude of parameter v_k itself. Regarding the winning ticket found via our IMS, the set of parameters selected in FEDSELECT will permanently remain personalized (i.e., always kept local and not sent to server); this set will grow in size over the course of FL.

There are two additional considerations when designing an algorithm that selects parameters using our hypothesis for personalization while training during FL: (1) computational efficiency for subnetwork discovery and (2) a suitable mechanism for performing a local update on both u_k and v_k. A possible algorithm for selecting v_k for each communication round could involve performing federated averaging, followed by a gradient-based subnetwork search using a modified version of the IMS that computes m using ∇v_k. However, this would require performing $j \times L$ iterations of local training, which would incur an undesirable computational overhead during FL. Next, we discuss how FEDSELECT addresses both points (1) and (2).

FEDSELECT (Algorithm 1) takes the following as input: the set of clients C, an initial model θ_I, the number of communication rounds T, the personalization rate p, and the personalization limit α. We reuse the notation from Section 3 to refer to global and personalized parameters as either u and v, or positions in the client masks m_k equal to 0 or 1, respectively.

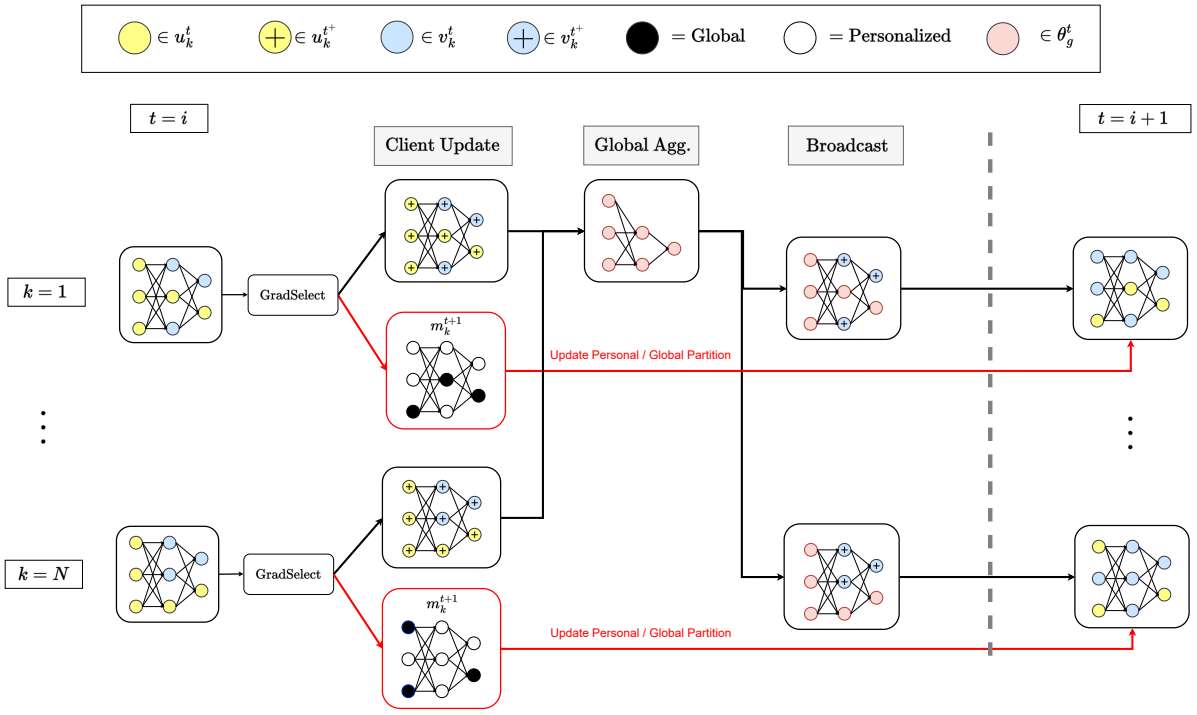

Figure 1. Illustration of the FEDSELECT algorithm. An example subnetwork update for communication round $t = i$ into $t = i + 1$ is depicted for N clients, where 2 clients are shown. There are 4 key steps: (1) the local update / new partition via GradSelect (Algorithm 3), (2) the aggregation of global parameters v_k^t, (3) broadcast of global parameters to the updated clients, and (4) application of the new mask m_k^{t+1} as a partition for the global/personalized parameters of each client in the subsequent round $t + 1$. In our algorithm, u_k^t denotes the global parameters for each client at round t; v_k denotes the personalized parameters for each client at round t; $u_k^{t^+}$ denotes the updated global parameters after GradSelect; $v_k^{t^+}$ denotes the updated personalized parameters after GradSelect; m_k^{t+1} is the binary mask with "0" denoting global parameter and "1" denoting personalized parameter; θ_g^t is the aggregate global parameters materialized each round.

4.2. Client Update Overview

At the beginning of FL, all client models are initialized with the same global model θ_g^0, denoted via binary client masks set as $m_k^0 = 0$ applied to each client model θ_k^0. Here, we use GradSelect as a black-box procedure for obtaining a personalized subnetwork update, which will be described in further detail in Section 4.5. First, given the current communication round t, GradSelect updates each clients' current global parameters (u_k^t) and personalized parameters (v_k^t); these are identified using the current mask m_k^t as an index into θ_k^t. Then, the next set of parameters to include for personalization in v_k^t are chosen according to a scalar personalization rate $p \in [0, 1]$. This is notated via values at the corresponding indices for v_k^t in m_k^t being set to 1. Importantly, in each communication round, the new parameters in v_k^t are selected solely from the set of global parameters that have been aggregated such that $\text{Idx}(v_k^0) \in \text{Idx}(v_k^1) \in \ldots \in \text{Idx}(v_k^T)$. We note a subtlety in our formulation in Eq. 2 that differs from the original PFL objective proposed in [34]. Namely, in the original partial model personalization framework, it is expected that all client models share the same global parameters denoted as u^t. In FEDSELECT, clients may learn different personalized parameters v_k^t, resulting in *differing partitions of the shared global parameters* u_k^t.

4.3. Global Aggregation

After each client performs its update, the global parameters u_k for each client are aggregated. Since m_k is likely to be heterogeneous across clients in typical personalized FL settings, averaging the global parameters u_k for each client requires careful handling. This is because the locations of global parameters (values of 0 in m_k) may be non-overlapping for multiple parameter indices. In FEDSELECT, global averaging for a given index in the current global model θ_g^t only occurs across parameters u_k in each client for which the corresponding mask entry in $m_k = 0$. We use ω_t as a tool of implementation to store the number of clients contributing to each global parameter in θ_g^t element-wise. By this construction, different subsets of clients $c_k \in C$ can contribute to different global parameters in θ_g^t. We provide a visual description of this procedure in Figure 1. As a result, the first communication round in FEDSELECT where all clients masks are set to 0 performs pure federated averaging. A consequence of this formulation is that in the extreme case where global parameters u_k is com-

Algorithm 1 FedSelect

Input: $C = \{c_1, \ldots, c_N\}, \theta_I, T, L$
Server Executes:
  Initialize all client models $\{\theta_i^0\}_{i=1}^N$ with θ_I
  Initialize all client masks $\{m_i^0\}_{i=1}^N$ with 0s
  for each round t in $0, 1, \ldots, R-1$ **do**
    for each client $c_k \in C$ **in parallel do**
      # Executed locally on client c_k
      $u_k^t \leftarrow \theta_k^t \odot \neg m_k^t$
      $v_k^t \leftarrow \theta_k^t \odot m_k^t$
      $u_k^{t^+}, v_k^{t^+}, m_k^{t+1} \leftarrow$ GradSelect(u_k^t, v_k^t)
    # Averaging occurs only across clients where the mask is 0 for a given parameter's position
    $\theta_g^t \leftarrow \mathbf{0}$
    $\omega^t \leftarrow \mathbf{0}$
    for $k = 1$ **to** N **do**
      $\theta_g^t \odot \neg m_k \leftarrow (\theta_g^t \odot \neg m_k^t) + u_k^{t^+}$
      $\omega^t \odot \neg m_k \leftarrow (\omega^t \odot \neg m_k^t) + \neg m_k^t$
    $m_g^t \leftarrow$ Binary mask for $\omega^t \neq 0$
    $\theta_g^t \odot m_g^t \leftarrow \frac{\theta_g^t \odot m_g^t}{\omega^t \odot m_g^t}$
    for $k = 1$ **to** N **do**
      # Distribute global params to clients' non-selected params, located via $\neg m_k^t$
      $\theta_k^{t+1} \odot \neg m_k^t \leftarrow \theta_g^t \odot \neg m_k^t$
      $\theta_k^{t+1} \odot m_k^t \leftarrow v_k^{t^+}$

Algorithm 2 LocalAlt$(u_{k,0}, v_{k,0})$ [34]

Input: Global/personalized parameters $u_{k,0}, v_{k,0}$, # of steps τ, global/local learning rates γ_v, γ_u, Batched data $\mathcal{D} = \{b_0, b_2, \ldots, b_{\tau-1}\}$
for $i = 0, 1, \ldots, \tau - 1$ **do**
  $v_{k,i+1} \leftarrow v_{k,i} - \gamma_v \nabla_{v_k} f_k((u_k, v_{k,i}), b_i)$
$v_k^+ \leftarrow v_{k,\tau}$
for $i = 0, 1, \ldots, \tau - 1$ **do**
  $u_{k,i+1} \leftarrow u_{k,i} - \gamma_u \nabla_{u_k} f_k((u_k, v_k^+), b_i)$
$u_k^+ \leftarrow u_{k,\tau}$
Return v_k^+, u_k^+

Algorithm 3 GradSelect(u_k, v_k)

Input: Global/personalized parameters u_k, v_k, Per. rate p, # local epochs L, Per. bound α
$u_{k,0} \leftarrow u_k$
$v_{k,0} \leftarrow v_k$
for $i = 0, 1, \ldots, L - 1$ **do**
  $u_{k,i+1}, v_{k,i+1} \leftarrow$ LocalAlt$(u_{k,i}, v_{k,i})$
if sparsity of $\neg m_k < \alpha$ **then**
  $\Delta u_k \leftarrow |u_{k,L} - u_{k,0}|$
  $m_k^+ \leftarrow$ binary mask for largest $p\%$ values in Δu_k
  # Element-wise Binary OR of m_k^+ and m_k
  $m_k^+ \leftarrow m_k^+ \vee m_k$
else
  $m_k^+ \leftarrow m_k$
Return $u_{k,L}, v_{k,L}, m_k^+$

pletely disjoint across all clients, each client will effectively undergo purely local training. However, full disjointedness across u_k is unlikely in typical FL settings where clients are likely to have similar data distributions. We hypothesize that clients with similar data distribution trained using FEDSELECT will learn similar personalized subnetworks; we show visualizations of these correlations in Section 5.2.

4.4. Subnetwork Representation

A key property of FEDSELECT, mentioned in Section 4.1, is that the personalized subnetwork representation gradually grows in size as the communication rounds progress. Specifically, the subnetwork size grows by a factor of $p\%$ until the corresponding client mask m_k reaches the maximum subnetwork mask sparsity α, a scalar defined within $[0, 1]$. We also refer to α as the personalization limit, the central hyperparameter of our proposed algorithm, where larger α indicates greater personalization. Therefore, when $\alpha = 1.0$, FEDSELECT computes local-only training of each client after m_k for each client reaches sparsity α. Conversely, when $\alpha = 0$, FEDSELECT reduces exactly to FEDAVG. We consider the behavior of our proposed algorithm as enabling a "rough interpolation" between these two extremes for personalization.

Furthermore, for $\alpha > 0.50$ and sufficiently many communication rounds for the chosen personalization rate p, a majority of parameters in all client models will be selected for personalization, resulting in decreased communication costs over time within FedSelect.

By our design of evolving masks over the course of FL rather than within a single communication round, FEDSELECT achieves a similar time complexity to other alternating minimization-based personalized FL methods like FedRep [7] and FedPAC [44], while performing a first-of-its-kind selection of personalized parameters.

4.5. GradSelect

We now describe GradSelect in isolation. GradSelect takes as input a partition of the current global parameters u_k^t and personalized parameters v_k^t at communication round t, a personalization rate p, and personalization limit α. The goal is to compute an update to the client model that grows the current subnetwork by a factor of $p\%$ while also training the current client model. To this end, we apply our FL Gradient-based Lottery Ticket Hypothesis to both train the client model's parameters and discover new parameters

for personalization. First, to update both u_k and v_k for each client, we use LocalAlt (Algorithm 2) from the partial model personalization framework [34]. LocalAlt was introduced to update a predefined set of shared and personalized parameters, u_k and v_k, by alternating full passes of stochastic gradient descent between both sets of parameters.

The LocalAlt training epochs within GradSelect are analogous to the training iterations used to compute lottery ticket updates in the aforementioned IMS procedure. Continuing the analogy, we store the state of the client parameters before LocalAlt θ_k, and compute the absolute value of the change of all global parameters after LocalAlt in Δu_k. Taking the largest $p\%$ of values in Δu_k, we create a the new mask m_k^+, which finishes the subnetwork update. Because m_k^+ is the mask to be used for local updates in the subsequent communication round, GradSelect also returns the newly-trained global/personalized parameters from the current mask partition m_k given by $u_{k,L}, v_{k,L}$. The parameter p controls the rate at which personalized subnetworks grow, and α controls the maximum size of the subnetwork. We include the result of varying α in Section 5.2, as well as the effect of changing p in Appendix B.

5. Experiments

In this section, we compare FedSelect with different approaches from FL and personalized FL. We use a variety of datasets and learning setups to demonstrate the superiority of FedSelect. We will make our source code publicly available to encourage reproducibility. Additional experimental results and details are provided in Appendix B.

5.1. Experimental Setup

Models & Datasets. We show results for our experimental setting across a breadth of benchmark datasets: CIFAR-10 [19], CIFAR10-C [15], Mini-ImageNet [42], and the OfficeHome dataset [41]. These settings additionally cover label shifts and feature shifts, which are common distributional shifts in real-world data. We use a ResNet18 [14] backbone with random initialization on CIFAR-10, CIFAR10-C, and Mini-ImageNet. For OfficeHome, we follow [39] to use a ResNet18 backbone pretrained on ImageNet [8].

Baselines. We compare our method to the *full model personalization* methods, including local-only training, FedAvg [31] with local fine-tuning (FedAvg + FT), Ditto [25], as well as *partial model personalization* methods, including FedPAC [44], FedBABU [33], FedRep [7], FedPer [5], and LG-FedAvg [27].

FL Settings & Hyperparameters. We consider the typical cross-silo setting [28] with dozens of clients and full-client participation. For the CIFAR-10 and CIFAR-10C experiments, the number of clients $N = |C| = 10$, with each client assigned $s = 2$ classes. In the Mini-ImageNet experiment, we set $N = 20$, with $s = 10$ classes assigned to each client. Finally, the OfficeHome experiment involves $N = 4$ clients, one for each of the 4 domain shifts in the dataset; each client is allocated all 65 classes. We vary the personalization limit, α, within $[0.05, 0.30, 0.50, 0.80]$. Further details on hyperparameter tuning for both FEDSELECT and the compared baselines are provided in Appendix A.

Method	CIFAR10	CIFAR10-C	Mini-ImageNet	Officehome
Local Only	74.60	69.50	34.98	75.60
FedAvg	27.70	21.35	8.06	74.35
FedAvg + FT	75.30	66.65	30.78	76.84
Ditto	72.75	63.70	32.30	77.99
FedPAC	77.20	69.50	37.72	66.61
FedRep	67.60	62.50	34.71	80.07
FedPer	75.40	65.95	24.90	69.01
FedBABU	75.45	64.95	6.99	65.63
LG-FedAvg	77.65	68.55	36.68	78.75
FedSelect	**82.25**	**72.05**	**38.69**	**80.51**

Table 1. Personalized accuracy of different methods on four datasets. FEDSELECT achieves the highest personalized performance.

Evaluation Metric. For all methods, the mean accuracy of the final model(s) across individual client data distributions calculated at the final communication round is reported. For FedAvg, accuracy is reported for a single global model. However, for other methods that learn personalized client models, the final average accuracy is reported by averaging individual clients' personalized model accuracies.

Training Settings. We use standard SGD for performing local client optimization for all methods based on their respective training objective. To fairly compare the performance of these methods, we fix the number of local training epochs across all methods to 3. The number of communication rounds T is set to 200 for all experiments except OfficeHome, where $T = 30$. For both CIFAR-10 and CIFAR10-C, each client is given 100 training samples. For Mini-ImageNet, we use 20% of the total training data, sampled for 20 clients in a non-iid manner. The 4 clients in the OfficeHome experiment were given the full training partition for each of their corresponding domains, resulting in about $2,000$ training samples per client. Further details on our data partition are provided in Appendix A.

5.2. Results

Personalization Performance. We report the results on four datasets in Table 1. In all cases, FEDSELECT achieves state-of-the-art results, surpassing baseline methods with an average improvement of over 2.4%. It is also clear that

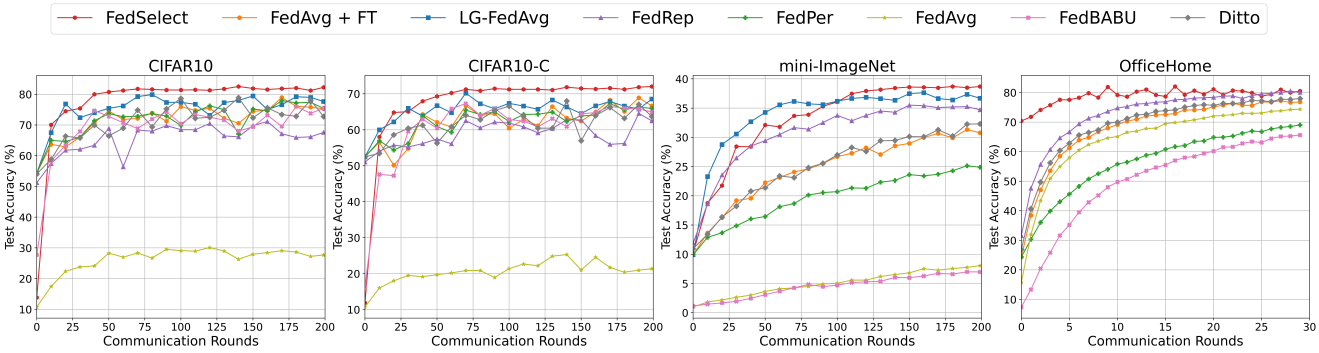

Figure 2. Test accuracy across communication rounds of FEDSELECT and baselines under the experimental settings in Table 1. FEDSELECT outperforms all baselines and exhibits more stable convergence.

Method	CIFAR10	CIFAR10-C	Mini-ImageNet	Officehome
FedSelect ($\alpha = 0.05$)	80.10	68.30	37.60	78.46
FedSelect ($\alpha = 0.30$)	**82.25**	70.60	37.84	**80.51**
FedSelect ($\alpha = 0.50$)	82.20	**72.05**	**38.69**	76.08
FedSelect ($\alpha = 0.80$)	81.40	71.40	34.51	76.65

Table 2. Performance of FEDSELECT when varying the personalization limit α.

Variant	CIFAR-10	CIFAR10-C	Mini-ImageNet	OfficeHome
Personalize Least	79.87	64.20	35.65	78.23
Layer A	79.96	65.76	35.43	77.06
Layer B	78.36	62.81	32.74	79.08
Layer C	79.24	64.82	33.33	79.64
Layer D	78.08	62.50	34.91	79.11
Random	79.73	61.12	33.28	76.33
FedSelect ($\alpha = 0.30$)	**82.25**	70.60	37.84	**80.51**
FedSelect ($\alpha = 0.50$)	82.20	**72.05**	**38.69**	76.08

Table 3. Ablation study for three variants: Personalize Least refers to the inverse of our hypothesis (personalize parameters with the least variation); Layer A/B/C/D refers to personalizing specific internal layers of ResNet18; Random refers to choosing a random partition between global and personal parameters.

our method is scalable to various feature/label shifts, evidenced by consistent performance across CIFAR10-C and Mini-ImageNet experiments. In contrast, the performance of methods such as FedBABU, FedPer, FedRep, and Ditto degrades significantly for these label/feature-shift experiments. We also note that FedAvg with fine-tuning (FedAvg + FT) performs competitively with other PFL baselines, which has also been previously observed in [7, 44]. The consistent superiority of FEDSELECT demonstrates the benefits of learning *which* parameters to personalize, *while* fine-tuning them. We also observe in the test accuracy convergence plots in Figure 2 that our method converges more smoothly to its final test accuracy than the other baselines. Early convergence in FL is useful for enabling early stopping and preventing further communication costs.

Effect of Personalization Limit α. In Table 2, we find that adjusting the personalization limit α enables the performance of FEDSELECT to be tuned under different client

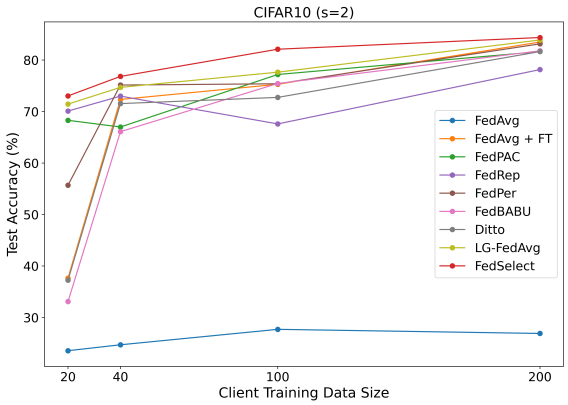

Figure 3. Personalized performance on CIFAR-10 with different local training data size and shard $s = 2$. FEDSELECT outperforms prior methods.

data distributions. Recall from Section 4.4 that the two extremes of FEDSELECT, $\alpha = 0$ and $\alpha = 1.0$, perform FedAvg and eventual local training, respectively. The results in Table 2 suggest that choosing a middle-ground between personalizing and globally averaging parameters is beneficial. In particular, we recommend $\alpha \in [0.3, 0.5]$ as suitable for most heterogenous client distributions. We present results for other values of α in Appendix B.

Effect of Training Data Size. We conduct an additional set of experiments on CIFAR-10 to demonstrate the performance of FEDSELECT and the aforementioned baseline methods as the size of training data increases. The purpose of this experiment is to stress-test the performance of the algorithms under significantly limited data settings, as well as consistency in performance as the client training sets scale in size. The results shown in Figure 3 showcase the robustness of FEDSELECT to a setting with significantly fewer data samples, whereas other baselines like FedBABU and FedPer degrade to as little as 33.1% and 55.7%, respectively. Full table results for this experiment are included in Appendix B.

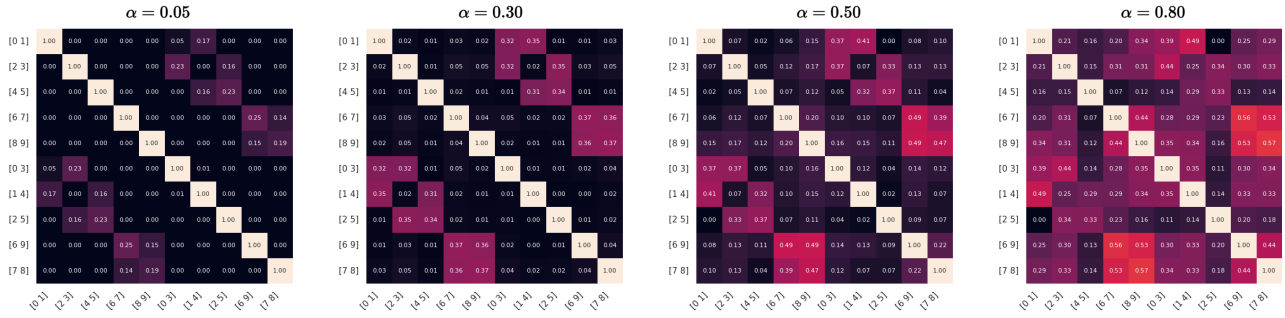

Figure 4. Normalized intersection-over-union (IoU) overlap of the subnetwork masks m_k in the final round in the ResNet18 final linear layer for each client in the CIFAR10 experiments from Table 2. Increasing α is shown from left-to-right. Each client was assigned 2 classes; the class labels are shown along the rows and columns of each matrix. Clients with similar labels develop similar subnetworks; increasing α results in more personalized parameters, but less distinct subnetworks.

Ablation Study. We perform ablation experiments to verify the design choices behind our proposed algorithm. Depicted in Table 3, we test the following three variants of parameter-wise FL algorithms. First, we address the inverse of our hypothesis, which is to personalize parameters that change the least (Personalize Least). Second, we test 4 different layers (denoted A, B, C, & D) in ResNet18 for individual personalization. In the case of the layer-wise study, we also note that baseline algorithms like FedRep and FedBABU are versions of parameter-wise PFL where the final linear layer is decoupled. We provide further details on the set of layers studied in Appendix B. Third, we test a random partition of personal vs. global parameters. In Table 3, we observe that all ablations perform worse than FEDSELECT, which further supports our FL Gradient-based Lottery Ticket Hypothesis.

Subnetwork Development During FL. While raw test accuracy of the final communication rounds provides useful insights into the performance of FEDSELECT, we also seek to visualize the behavior of the clients' collaboration of both parameters and subnetworks. In Figure 4, we showcase the similarity of subnetwork structures across clients based on their label distributions. We observe that clients that share at least one label have exhibit significant overlap in their subnetwork masks. We also note that the increased overlap in subnetwork parameters due to increasing α results in less distinct but more locally trained parameters.

6. Limitations

In this work, we mainly focus on improving the personalization performance of FL. Nevertheless, personalized federated learning faces challenges in balancing personalization and model generalization (e.g., test-time distribution shifts), as local updates may lead to overfitting and biased outcomes. Heterogeneous datasets contribute to diverse knowledge for FL, while security risks such as adversarial attacks and model poisoning persist. The decentralized nature introduces communication overhead and resource demands, impacting scalability and real-time responsiveness. Ongoing research is crucial to address these limitations and strike a balance between personalization, efficiency, model robustness, and privacy in federated learning systems.

7. Conclusion

In this work, we propose FedSelect, an approach that adaptively selects model parameters for personalization while concurrently conducting global aggregations on the remaining parameters in personalized federated learning. FedSelect represents a significant stride towards achieving a harmonious balance between individual customization and collective model performance while reducing communication costs. By dynamically tailoring specific parameters to local data characteristics, FedSelect mitigates the risk of overfitting and enhances personalization. Simultaneously, its global aggregation mechanism ensures the model maintains robust and generalized performance across the entire federated network. Finally, we evaluated FedSelect on multiple datasets with different learning setups and showed that it outperforms previous approaches by a significant margin. The impressive performance of FedSelect paves the way for intriguing future research directions in this domain.

Acknowledgements. We thank Bo Li for the initial discussion and constructive suggestions. We also thank the National Center for Supercomputing Applications (NCSA) and Illinois Campus Cluster Program (ICCP) for supporting our computing needs. This work used NVIDIA GPUs at NCSA Delta through allocations CIS230117 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by NSF Grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. 2

[2] Idan Achituve, Aviv Shamsian, Aviv Navon, Gal Chechik, and Ethan Fetaya. Personalized federated learning with gaussian processes. *Advances in Neural Information Processing Systems*, 34:8392–8406, 2021. 2

[3] Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *ArXiv*, abs/2003.12880, 2020. 2

[4] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8: 140699–140725, 2020. 2

[5] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019. 1, 3, 6

[6] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication, 2019. 2

[7] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pages 2089–2099. PMLR, 2021. 1, 2, 3, 5, 6, 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[9] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33:2661–2674, 2021. 2

[10] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach, 2020. 2

[11] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. 1, 3

[12] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis, 2020. 1

[13] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68: 8076–8091, 2020. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 6

[16] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution, 2020. 2

[17] Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning, 2023. 2

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 6

[20] Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts, 2023. 3

[21] 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, 2020. 2, 3

[22] Q. Li, Zeyi Wen, Zhaomin Wu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 35:3347–3366, 2019. 2

[23] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1

[24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020. 2

[25] T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 6

[26] 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. 3

[27] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1, 3, 6

[28] Ziyu Liu, Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. On privacy and personalization in cross-silo federated learning, 2022. 6

[29] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning, 2020. 2

[30] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989. 2

[31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. of Int'l Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017. 1, 2, 6

[32] Vaikkunth Mugunthan, Eric Lin, Vignesh Gokul, Christian Lau, Lalana Kagal, and Steve Pieper. Fedltn: Federated learning for sparse and personalized lottery ticket networks. In *Computer Vision – ECCV 2022*, pages 69–85, Cham, 2022. Springer Nature Switzerland. 2

[33] Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. 1, 6

[34] K. Pillutla, K. Malik, A. Mohamed, M. Rabbat, M. Sanjabi, and L. Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, 2022. 2, 3, 4, 5, 6

[35] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning, 2020. 1

[36] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021. 2

[37] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, and S. Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(12598), 2020. 1

[38] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *ArXiv*, abs/1705.10467, 2017. 2

[39] Benyuan Sun, Hongxing Huo, Yi Yang, and Bo Bai. Partialfed: Cross-domain personalized federated learning via partial initialization. *Advances in Neural Information Processing Systems*, 34:23309–23320, 2021. 6

[40] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[41] Hemanth Venkateswara, José Eusébio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[42] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Neural Information Processing Systems*, 2017. 6

[43] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning, 2020. 2

[44] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 5, 6, 7

[45] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Neural Information Processing Systems*, 2014. 3

[46] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation, 2022. 2

[47] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. 2018. 2