FedSSA: Semantic Similarity-based Aggregation for Efficient Model-Heterogeneous Personalized Federated Learning

Liping Yi[1,2]**, Han Yu**[2]**, Zhuan Shi**[3]**, Gang Wang**[1,*]**, Xiaoguang Liu**[1]**, Lizhen Cui**[4,*]**, Xiaoxiao Li**[5]

[1]College of Computer Science, TMCC, SysNet, DISSec, GTIISC, Nankai University, Tianjin, China

[2]College of Computing and Data Science, Nanyang Technological University, Singapore

[3]Artificial Intelligence Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

[4]School of Software, Shandong University, Jinan, China

[5]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada

{yiliping, wgzwp, liuxg}@nbjl.nankai.edu.cn, han.yu@ntu.edu.sg, zhuan.shi@epfl.ch, clz@sdu.edu.cn, xiaoxiao.li@ece.ubc.ca

Abstract

Federated learning (FL) is a privacy-preserving collaboratively machine learning paradigm. Traditional FL requires all data owners (a.k.a. FL clients) to train the same local model. This design is not well-suited for scenarios involving data and/or system heterogeneity. Model-Heterogeneous Personalized FL (MHPFL) has emerged to address this challenge. Existing MHPFL approaches often rely on a public dataset with the same nature as the learning task, or incur high computation and communication costs. To address these limitations, we propose the Federated Semantic Similarity Aggregation (FedSSA) approach for supervised classification tasks, which splits each client's model into a heterogeneous (structure-different) feature extractor and a homogeneous (structure-same) classification header. It performs local-to-global knowledge transfer via semantic similarity-based header parameter aggregation. In addition, global-to-local knowledge transfer is achieved via an adaptive parameter stabilization strategy which fuses the seen-class parameters of historical local headers with that of the latest global header for each client. FedSSA does not rely on public datasets, while only requiring partial header parameter transmission to save costs. Theoretical analysis proves the convergence of FedSSA. Extensive experiments present that FedSSA achieves up to 3.62% higher accuracy, 15.54 times higher communication efficiency, and 15.52 times higher computational efficiency compared to 7 state-of-the-art MHPFL baselines.

1 Introduction

As societies become increasingly aware of the importance of data privacy protection, centrally collecting large-scale data
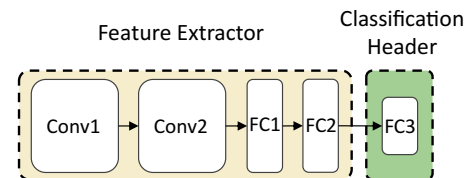
*Corresponding authors.

Figure 1: Feature extractor and classification header.

for model training has become less viable, especially under privacy regulations. To enable decentralized data owners to collaboratively train effective machine learning models while protecting privacy, federated learning (FL) [McMahan *et al.*, 2017] has been proposed. In a typical FL system, a central FL server coordinates multiple data owners (FL clients) for model training. In each communication round, the server broadcasts its current global model to the clients. Each client then treats the received global model as its local model, and trains it on local private data. The updated local model is uploaded to the server. The server weighted averages (aggregates) the received local models to obtain an updated global model. The above steps are repeated until the global model converges. In FL, only model parameters are communicated between the server and clients. Potentially sensitive local data are not exposed, thereby protecting privacy [Liu *et al.*, 2024].

This model homogeneous mode of FL requires all clients to train models with the same structures. It is not well-suited for scenarios involving data heterogeneity [Zhu *et al.*, 2021] (i.e., clients with local data following non-independent and identical distributions (non-IID)) and system heterogeneity [Jiang *et al.*, 2022; Yi *et al.*, 2024b; Yu *et al.*, 2011; Zhang *et al.*, 2023] (i.e., clients with diverse communication and computation capabilities). Therefore, training a personalized FL model adaptively based on each client's local data distribution and system capability helps improve performances [Tan *et al.*, 2022a; Yi *et al.*, 2024a]. Furthermore, enterprise FL clients are often reluctant to share their model structures with others due to intellectual property concerns. Thus, the field of Model-Heterogeneous Personalized Federated Learning (MHPFL) [Yi *et al.*, 2023a; Yi *et al.*, 2023c;

Yi *et al.*, 2023b; Yi *et al.*, 2024c], which allows each client to train a heterogeneous personalized local model with a structure tailored to actual needs, has emerged.

Existing MHPFL methods are mainly designed based on knowledge distillation, mutual learning, and model mixup. Knowledge distillation-based methods [Li and Wang, 2019; Lin *et al.*, 2020; Huang *et al.*, 2022a; Itahara *et al.*, 2023; Sattler *et al.*, 2022; Makhija *et al.*, 2022; Chang *et al.*, 2021] often rely on the availability of a public dataset that is large enough and follows a similar distribution with local private data for fusing knowledge from heterogeneous local models. However, such a public dataset is difficult to obtain in practice. Other knowledge distillation-based methods [Jeong *et al.*, 2018; Ahn *et al.*, 2019; Ahn *et al.*, 2020; Tan *et al.*, 2022b] without requiring public datasets often incur high computational overhead on clients and face the risk of privacy leakage. Mutual learning-based methods [Shen *et al.*, 2023; Wu *et al.*, 2022] allow each client to train a large heterogeneous model and a small homogeneous model in a mutual learning manner. Only the small homogeneous models are uploaded to the server for aggregation. Training two models simultaneously leads to high computational overhead for clients. Model mixup-based methods [Collins *et al.*, 2021; Liang *et al.*, 2020; Jang *et al.*, 2022] divide each client's local model into a heterogeneous feature extractor and a homogeneous classifier. The server aggregates the homogeneous local classifiers to update the global classifier and then broadcasts it to clients. Each client replaces its local classifier with the updated global classifier or constructs a regularization term based on the distillation loss between local and global classifiers. They often ignore the semantic information of the classifier parameters, resulting in limited accuracy improvements.

To enable the privacy-preserving, communication- and computation-efficient training of high-performance models through MHPFL, we propose the Federated Semantic Similarity Aggregation (FedSSA) approach for supervised classification tasks, which have been widely applied in fields like cancer detection for medical diagnosis and object recognition in autonomous driving [Kairouz *et al.*, 2021]. As depicted in Figure 1, it splits each client's local model into two parts: 1) a heterogeneous feature extractor, and 2) a homogeneous classification header. Local-to-global knowledge transfer is achieved by semantic similarity-based classification header parameter aggregation. Since the parameters of classification headers from different clients corresponding to the same class are semantically similar, each client only needs to upload classification header parameters corresponding to locally seen classes. The FL server aggregates parameters from different classification headers by class to update the global header. Global-to-local knowledge transfer is achieved through parameter stabilization. To ensure stable convergence, we devise an adaptive parameter stabilization strategy that fuses local historical header parameters and the latest global header parameters for locally seen classes to update the local header of each FL client.

Owing to the above design, FedSSA has the following advantages: 1) aggregating semantically similar head parameters corresponding to the same class stabilizes the decision boundaries and boosts performance; 2) the adaptive parameter stabilization strategy alleviates parameter shaking in the initial training rounds, thereby speeding up convergence; 3) the server and clients only need to transmit classification header parameters corresponding to seen classes, which is communication-efficient compared to transmitting the entire model; 4) low computational overhead for FL clients as they only fuse partial parameters from the global and local headers before local training; and 5) since only partial parameters are transmitted between the FL server and FL clients, no new privacy risk is introduced. In short, compared with existing methods, FedSSA improves model performance, and communication and computation efficiency simultaneously.

Theoretical analysis proves the convergence of FedSSA. Extensive experiments on two real-world datasets against seven compared with state-of-the-art MHPFL baselines demonstrate that FedSSA achieves up to 3.62% higher accuracy, 15.54 times higher communication efficiency, and 15.52 times higher computation efficiency.

2 Related Work

Our work is closely related to MHPFL with *complete* model heterogeneity. These methods support flexible model heterogeneity, and are divided into the following three categories.

Knowledge Distillation-based MHPFL. Most knowledge distillation-based MHPFL methods rely on a public dataset (e.g., FedMD [Li and Wang, 2019], FedDF [Lin *et al.*, 2020], FCCL [Huang *et al.*, 2022a], DS-FL [Itahara *et al.*, 2023], CFD [Sattler *et al.*, 2022], FedHeNN [Makhija *et al.*, 2022], Cronus [Chang *et al.*, 2021], FSFL [Huang *et al.*, 2022b], FedAUX [Sattler *et al.*, 2021], FedKT [Li *et al.*, 2021], Fed-ET [Cho *et al.*, 2022], FedKEMF [Yu *et al.*, 2022], FedGEMS [Cheng *et al.*, 2021], KT-pFL [Zhang *et al.*, 2021]) to fuse information of local heterogeneous models by knowledge distillation. However, the public dataset is not always available. To avoid relying public datasets, FedGen [Zhu and others, 2021] and FedZKT [Zhu *et al.*, 2022] train a generator for generating local representations or public shared datasets, which is time-consuming and computation-intensive. In FD [Jeong *et al.*, 2018], HFD [Ahn *et al.*, 2019; Ahn *et al.*, 2020], FedProto [Tan *et al.*, 2022b], FedGKT [He *et al.*, 2020], each client uploads the (class-average) logits or representations of local data to the server, the aggregated global logits or representations for each class are sent to clients for local knowledge distillation which incurs high computational costs for clients. Besides, uploading the logits/representations might compromise privacy.

Mutual Learning-based MHPFL. FML [Shen *et al.*, 2023] and FedKD [Wu *et al.*, 2022] enable each client to train a large heterogeneous model and a small homogeneous model in a mutual learning manner. The large model is always trained locally and the small model is uploaded to the server for aggregation. Although they implement information interaction through the small homogeneous models, training the homogeneous model increases local computational costs for clients, and transmitting the homogeneous models incurs high communication costs.

Model Mixup-based MHPFL. These methods split each

local model into a feature extractor and a classifier. In `FedRep` [Collins *et al.*, 2021], `FedPer` [Arivazhagan *et al.*, 2019], `FedMatch` [Chen *et al.*, 2021], `FedBABU` [Oh *et al.*, 2022] and `FedAlt/FedSim` [Pillutla *et al.*, 2022], the feature extractor is homogeneous and used for aggregation by the FL server to enhance generalization. The classifier can be heterogeneous. Since the feature extractor has more parameters than the classifier, these methods can only support model heterogeneity to a low degree. In contrast, `LG-FedAvg` [Liang *et al.*, 2020], `FedClassAvg` [Jang *et al.*, 2022] and `CHFL` [Liu *et al.*, 2022] use heterogeneous feature extractors and homogeneous classifiers (i.e., executing the same classification task). The local classifiers are uploaded to the FL server for aggregation to generate the global classifier. To acquire global knowledge, `LG-FedAvg` directly replaces each client's local classifier with the global classifier. Each client in `CHFL` or `FedClassAvg` calculates the regularization term or distillation loss between the local and global classifiers. Although these methods support a higher degree of model heterogeneity, they ignore the semantic similarity of classifier parameters belonging to the same class, thus achieving limited performance improvement. Our `FedSSA` sets out to address the aforementioned limitations.

3 Preliminaries

3.1 Notations and Objective of Typical FL

A typical FL system involves a central FL server and N decentralized FL clients. In each round, the server selects a C fraction of clients at random. The selected client set is denoted as $\mathcal{K}$, $|\mathcal{K}| = C \cdot N = K$. The server then broadcasts the global model $f(\omega)$ ($f(\cdot)$ and ω denote the model structure and the model parameters) to the selected clients. A client k trains the received $f(\omega)$ on its local dataset D_k to produce a local model $f(\omega_k)$ through gradient descent $\omega_k \leftarrow \omega - \eta \nabla \ell(f(\boldsymbol{x}_i; \omega), y_i)$. $\ell(f(\boldsymbol{x}_i; \omega), y_i)$ is the loss of the global model $f(\omega)$ on the sample $(\boldsymbol{x}_i, y_i) \in D_k$. $D_k \sim P_k$ indicates that D_k obeys distribution P_k (i.e., local data from different clients are non-IID). Then, client k uploads its local model parameters ω_k to the server. The server aggregates the received local models to update the global model, $\omega = \sum_{k=0}^{K-1} \frac{n_k}{n} \omega_k$. That is, the objective of typical FL is to minimize the average loss of the global model $f(\omega)$ on data from all clients:

$$\min_{\omega \in \mathbb{R}^d} \sum_{k=0}^{K-1} \frac{n_k}{n} \mathcal{L}_k(D_k; f(\omega)), \qquad (1)$$

where ω are d-dimensional real numbers. $n_k = |D_k|$ is the number of samples in client k's local dataset. $n = \sum_{k=0}^{N-1} n_k$. $\mathcal{L}_k(D_k; f(\omega)) = \frac{1}{|D_k|} \sum_{(\boldsymbol{x}_i, y_i) \in D_k} \ell(f(\boldsymbol{x}_i; \omega), y_i)$ is the loss of the global model $f(\omega)$ on D_k.

3.2 Problem Definition

The problem we aim to solve in this work belongs to the category of MHPFL for *supervised classification* tasks. Each FL client k owns local models $f_k(\omega_k)$ with a model structure

$f_k(\cdot)$ and parameters ω_k. They can be heterogeneous for different clients. We assume all clients execute the same classification task and each client's local model $f_k(\omega_k)$ consists of a heterogeneous feature extractor $\mathcal{F}_k(\varphi_k)$ and a homogeneous classification header $\mathcal{H}(\theta_k)$, i.e., $f_k(\omega_k) = \mathcal{F}_k(\varphi_k) \circ \mathcal{H}(\theta_k)$ with $\circ$ denoting model splicing. The feature extractor takes inputs as $\mathcal{F}_k(D_k; \varphi_k)$, where $\mathcal{F}_k(\cdot), \varphi_k$ denote the structure and parameters of the feature extractor. It maps data D_k from the input space to the feature space. The classification header takes input as $\mathcal{H}(\mathcal{F}_k(D_k; \varphi_k); \theta_k)$, where $\mathcal{H}(\cdot), \theta_k$ denote the structure and parameters of the classification header. It involves the last two linear layers of the model which maps the features $\mathcal{F}_k(D_k; \varphi_k)$ extracted by $\mathcal{F}_k(\cdot)$ to the output space.

Our objective is to minimize the sum of losses of all clients' heterogeneous local models:

$$\min_{\omega_0, \ldots, \omega_{K-1}} \sum_{k=0}^{K-1} \mathcal{L}_k(D_k; f_k(\omega_k)) = \sum_{k=0}^{K-1} \mathcal{L}_k(D_k; \mathcal{F}_k(\varphi_k) \circ \mathcal{H}(\theta_k)), \qquad (2)$$

where local models $\omega_0, \ldots, \omega_{K-1} \in \mathbb{R}^{d_0, \ldots, d_{K-1}}$.

4 The Proposed `FedSSA` Approach

`FedSSA` consists of two modules: 1) semantic similarity-based classification header parameter aggregation for local-to-global knowledge transfer, and 2) adaptive parameter stabilization for global-to-local knowledge transfer.

4.1 Semantic Similarity-Based Aggregation

As shown in Figure 2, the last fully connected layer of the classification header outputs all-class logits which are mapped to the prediction probability for each class by a soft-max layer. That is, each neuron at the classification header's last layer corresponds to one class and its connections with all neurons at the previous layer are the parameters belonging to corresponding classes. Assuming that all clients execute the same S-classification task, then client k's local classification header θ_k can be subdivided by class as $\{\theta_k^0, \ldots, \theta_k^s, \ldots, \theta_k^{S-1}\}$. Since local data from different clients are non-IID, different clients may hold partial classes of data. We denote the set of client k's seen classes as S_k.
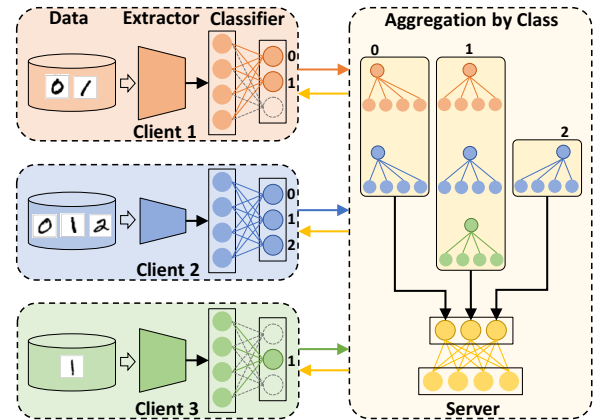

Figure 2: The `FedSSA` framework.

Since each client trains its local model on locally seen classes, the local classification header seen-class parameters are well-trained. Therefore, client k only uploads its classification header seen-class parameters $\{\theta_k^s\}, s \in S_k$ to the FL server for aggregation. As illustrated in Figure 2, client 1 only uploads its classification header parameters corresponding to seen classes '0, 1', client 2 only uploads its classification header parameters for classes '0, 1, 2', and client 3 only uploads its classification header parameters for class '1'.

Since classification header parameters for the same class from different clients are *semantically similar*, we design the server to aggregate classification header parameters based on classes (i.e., the corresponding neuron positions at the output layer). We denote the set of clients holding class s of data as $\mathcal{K}_s$, and define the aggregation rule as:

$$\theta^s = \frac{1}{|\mathcal{K}_s|} \sum_{k \in \mathcal{K}_s} \theta_k^s. \tag{3}$$

After aggregation, the server splices the aggregated header parameters for all classes together to form an updated global header $\theta = \{\theta^0, \ldots, \theta^s, \ldots, \theta^{S-1}\}$. Then, it sends the partial parameters of the global header to each client corresponding to its respective locally seen classes. This step transfers local seen-class knowledge from clients to the server. It enhances the generalization to seen classes for each client after receiving the aggregated seen-class header parameters, and produces a more stable seen-class classification boundary.

4.2 Adaptive Parameter Stabilization

Inspired by historical learning [Li *et al.*, 2023] which tends to re-use historical knowledge from old models to avoid forgetting, during global-to-local knowledge transfer, we exploit historical local headers to stabilize and speed up model convergence. Specifically, in the t-th training round, client k fuses the seen-class parameters from its historical local header $\theta_k^{t-1,s}$ ($s \in S_k$) and the latest global header $\theta^{t-1,s}$ ($s \in S_k$) to produce a new local header $\hat{\theta}_k^{t,s}$. The unseen-class header parameters are still the historical local header $\theta_k^{t-1,s'}$ ($s' \in (S \backslash S_k)$). The fusion of the seen-class parameters from the two headers is achieved by the proposed adaptive parameter stabilization strategy.

In the initial rounds of training, local model parameters are updated rapidly and local model parameters from different

Figure 3: Decay functions for μ^t. $\cos(\cdot) \in [0, \pi/2]$ is a smooth decay function. It leads to a stable decrease of μ.

clients trained on non-IID data are biased. This leads to instability in the global header. To enhance the reliability of global-to-local knowledge transfer, we devise an adaptive parameter stabilization strategy:

$$\hat{\theta}_k^{t,s} = \theta^{t-1,s} + \mu^t \cdot \theta_k^{t-1,s}, s \in S_k,$$

$$\mu^t = \begin{cases} \mu_0 \cdot \cos(\frac{t}{2T_{stable}}\pi), & t \leq T_{stable}, \\ 0, & t > T_{stable}. \end{cases} \tag{4}$$

The coefficient $\mu^t \in [0, 1]$ of the historical local header is determined by the above piecewise function. In round $t \leq T_{stable}$ (a fixed hyperparameter for all clients in FL), μ^t is determined by a decay function (e.g., Figure 3) with initial value $\mu_0 \in (0, 1]$. μ^t decreases as round t rises. When $t > T_{stable}$, the reliability of the global header has been enhanced. Thus, we set $\mu^t = 0$ to directly replace historical local seen-class parameters with the latest global seen-class header parameters. This prevents stateless local header parameters from slowing down convergence.

The updated local header $\hat{\theta}_k^t = \{\hat{\theta}_k^{t,s}, \theta_k^{t-1,s'}\}(s \in S_k, s' \in (S \backslash S_k))$ and the feature extractor φ_k^{t-1} from the previous round are spliced together to form a new local model $\hat{\omega}_k^t$:

$$\hat{\omega}_k^t = \varphi_k^{t-1} \circ \hat{\theta}_k^t. \tag{5}$$

Then, $\hat{\omega}_k^t$ is trained on D_k to obtain ω_k^t via gradient descent:

$$\omega_k^t \leftarrow \hat{\omega}_k^t - \eta \nabla(f_k(\boldsymbol{x}_i; \hat{\omega}_k^t), y_i), (\boldsymbol{x}_i, y_i) \in D_k. \tag{6}$$

These three key steps of *uploading seen-class header parameters by class*, *aggregation by class*, and *parameter fusion by class* enhance the personalization and stabilize the classification boundary for each heterogeneous local model. They are repeated until local models from all clients converge. Each client's final local model is used for inference. `FedSSA` is detailed in Algorithm 1 (Appendix A [1]).

4.3 Discussion

Here, we discuss the privacy, communication costs, and computational overheads of `FedSSA`.

Privacy. When a client uploads its seen-class header parameters, it can maintain the seen-class header parameters and replace the unseen-class header parameters with 0, while still uploading the entire header to the server. The server aggregates non-zero local header parameters according to the corresponding classes (ordinates), and then broadcasts the updated global header to the clients. Since only the header parameters are transmitted, local data privacy can be preserved.

Communication Cost. As stated above, only the parameters of headers are communicated between clients and the server in each round of FL. Therefore, `FedSSA` incurs lower communication costs than transmitting the complete models as in the cases of `FedAvg` based FL approaches.

Computational overhead. For clients, the additional computation compared with `FedAvg` is incurred by header parameter fusion in the adaptive parameter stabilization strategy, which are simple linear operations. It is significantly
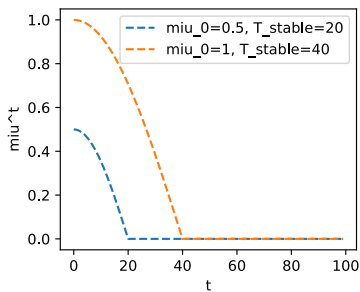
---

[1]The appendices of this paper can be accessible from https://arxiv.org/abs/2312.09006

lower than the computational overhead incurred by training local models. For the server, aggregating the header parameters by class linearly incurs low computation costs.

5 Analysis

To analyze the convergence of FedSSA, we first introduce some additional notations. Let $e \in \{0, 1, ..., E\}$ denote a local iteration. $(tE + e)$ denotes the e-th local iteration in the FL training round $(t + 1)$. At $(tE + 0)$ (i.e., the beginning of the round $(t + 1)$), clients fuse historical local headers (in round t) and the latest global header (in round t) to update local headers. $(tE + 1)$ is the first iteration in round $(t + 1)$. $(tE + E)$ is the last iteration in round $(t + 1)$.

Assumption 1. *Lipschitz Smoothness. Client k's local model gradients are $L1$–Lipschitz smooth, i.e.,*

$$\|\nabla \mathcal{L}_k^{t_1}(\omega_k^{t_1}; \boldsymbol{x}, y) - \nabla \mathcal{L}_k^{t_2}(\omega_k^{t_2}; \boldsymbol{x}, y)\| \leqslant L_1 \|\omega_k^{t_1} - \omega_k^{t_2}\|, \\ \forall t_1, t_2 > 0, k \in \{0, 1, \ldots, N-1\}, (\boldsymbol{x}, y) \in D_k. \quad (7)$$

From Eq. (7), we further derive:

$$\mathcal{L}_k^{t_1} - \mathcal{L}_k^{t_2} \leqslant \langle \nabla \mathcal{L}_k^{t_2}, (\omega_k^{t_1} - \omega_k^{t_2}) \rangle + \frac{L_1}{2} \|\omega_k^{t_1} - \omega_k^{t_2}\|_2^2. \quad (8)$$

Assumption 2. *Unbiased Gradient and Bounded Variance. The random gradient $g_k^t = \nabla \mathcal{L}_k^t(\omega_k^t; \mathcal{B}_k^t)$ ($\mathcal{B}$ is a batch of local data) of each client's local model is unbiased, i.e.,*

$$\mathbb{E}_{\mathcal{B}_k^t \subseteq D_k}[g_k^t] = \nabla \mathcal{L}_k^t(\omega_k^t), \quad (9)$$

and the variance of random gradient g_k^t is bounded by:

$$\mathbb{E}_{\mathcal{B}_k^t \subseteq D_k}[\|\nabla \mathcal{L}_k^t(\omega_k^t; \mathcal{B}_k^t) - \nabla \mathcal{L}_k^t(\omega_k^t)\|_2^2] \leqslant \sigma^2. \quad (10)$$

Assumption 3. *Bounded Variance of Classification Headers. The variance between the seen-class parameters $\theta_k^s (s \in S_k)$ of client k's local header $\mathcal{H}(\theta_k)$ and the same seen-class parameters $\theta^s = \mathbb{E}_{k' \in \mathcal{K}_s}[\theta_{k'}^s](s \in S_k)$ of the global header $\mathcal{H}(\theta)$ is bounded [Yi et al., 2023a]:*
Parameter bounded: $\mathbb{E}_{s \in S_k}[\|\theta^s - \theta_k^s\|_2^2] \leq \varepsilon^2$,
Gradient bounded: $\mathbb{E}_{s \in S_k}[\|\nabla \mathcal{L}(\theta^s) - \nabla \mathcal{L}(\theta_k^s)\|_2^2] \leq \delta_1^2$.

With the above assumptions, since FedSSA does not change local model training, Lemma 1 from [Tan *et al.*, 2022b] still holds.

Lemma 1. *Based on Assumptions 1 and 2, during $\{0, 1, ..., E\}$ iterations of the $(t + 1)$-th FL training round, the loss of an arbitrary client's local model is bounded by:*

$$\mathbb{E}[\mathcal{L}_{(t+1)E}] \leqslant \mathcal{L}_{tE+0} - (\eta - \frac{L_1 \eta^2}{2}) \sum_{e=0}^{E} \|\mathcal{L}_{tE+e}\|_2^2 + \frac{L_1 E \eta^2}{2} \sigma^2. \quad (11)$$

Lemma 2. *Based on Assumption 3, the loss of an arbitrary client's local model after fusing the seen-class parameters of local and global headers by the adaptive parameter stabilization strategy is bounded by:*

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leqslant \mathbb{E}[\mathcal{L}_{(t+1)E}] - \frac{\eta L_1 \delta^2}{2}. \quad (12)$$

The detailed proof can be found in Appendix B.

Based on Lemma 1 and Lemma 2, we can further derive the following theorems.

Theorem 1. *One-round deviation. The expectation of the loss of an arbitrary client's local model before the beginning of a round of local iteration satisfies*

$$\mathbb{E}[\mathcal{L}_{(t+1)E+0}] \leqslant \mathcal{L}_{tE+0} - (\eta - \frac{L_1 \eta^2}{2}) \sum_{e=0}^{E} \|\mathcal{L}_{tE+e}\|_2^2 \\ + \frac{\eta L_1 (E \eta \sigma^2 - \delta^2)}{2}. \quad (13)$$

The proof can be found in Appendix C. Then we can derive the non-convex convergence rate of FedSSA as follows:

Theorem 2. *Non-convex convergence rate of FedSSA. Based on the above assumptions and derivations, for an arbitrary client and any $\epsilon > 0$, the following inequality holds:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \sum_{e=0}^{E} \mathbb{E}[\|\mathcal{L}_{tE+e}\|_2^2] \leqslant \frac{2(\mathcal{L}_{t=0} - \mathcal{L}^*)}{T\eta(2 - L_1\eta)} + \frac{L_1(E\eta\sigma^2 - \delta^2)}{2 - L_1\eta} \leqslant \epsilon, \\ s.t. \ \eta < \frac{2\epsilon + L_1\delta^2}{L_1(\epsilon + E\sigma^2)}. \quad (14)$$

Therefore, under FedSSA, an arbitrary client's local model converges at a non-convex convergence rate of $\epsilon \sim \mathcal{O}(\frac{1}{T})$. The detailed proof can be found in Appendix D.

6 Experimental Evaluation

In this section, we compare FedSSA [2] against 7 state-of-the-art MHPFL approaches on two real-world datasets under various experiment conditions. and baselines with Pytorch on four NVIDIA GeForce RTX 3090 GPUs with 24G memory.

6.1 Experiment Setup

Datasets and Models. We evaluate FedSSA and baselines on two image classification datasets: CIFAR-10 and CIFAR-100 [3] [Krizhevsky, 2009]. They are manually divided into non-IID datasets following the method specified in [Shamsian *et al.*, 2021]. For CIFAR-10, we assign only data from 2 out of the 10 classes to each client (non-IID: 2/10). For CIFAR-100, we assign only data from 10 out of the 100 classes to each client (non-IID: 10/100). Then, each client's local data are further divided into the training set, the evaluation set, and the testing set following the ratio of 8:1:1. This way, the testing set is stored locally by each client, which follows the same distribution as the local training set. As shown in Table 3 (Appendix E), each client trains a CNN model with output layer dimensions (i.e., the last fully-connected layer) of 10 or 100 on CIFAR-10 and CIFAR-100 datasets, respectively. The dimensions of the representation layer (i.e., the second last fully-connected layer) are set to be 500. Therefore, each classification header consists of $10(100) \times 500$ parameters.

Baselines. We compare FedSSA with 7 baseline methods, which are the state-of-the-art methods in the three MHPFL categories elaborated in Sec. 2.

- Standalone, each client trains its local model independently, which serves as a lower performance bound;

---

[2]https://github.com/LipingYi/FedSSA
[3]https://www.cs.toronto.edu/%7Ekriz/cifar.html

	N=10, C=100%		N=50, C=20%		N=100, C=10%	
Method	CIFAR10	CIFAR100	CIFAR10	CIFAR100	CIFAR10	CIFAR100
Standalone	96.35	74.32	95.25	62.38	92.58	54.93
FML	94.83	70.02	93.18	57.56	87.93	46.20
FedKD	94.77	70.04	92.93	57.56	90.23	50.99
LG-FedAvg	96.47	73.43	94.20	61.77	90.25	46.64
FedClassAvg	96.44	74.33	94.45	63.10	91.03	49.31
FD	96.30	-	-	-	-	-
FedProto	95.83	72.79	95.10	62.55	91.19	54.01
FedSSA	**98.57**	**74.81**	**95.32**	**64.38**	**92.67**	**58.54**

Table 1: The mean test accuracy for *model-homogeneous* scenarios. N is the total number of clients. C is the fraction of participating clients in each round. '-' denotes failure to converge.

	N=10, C=100%		N=50, C=20%		N=100, C=10%	
Method	CIFAR10	CIFAR100	CIFAR10	CIFAR100	CIFAR10	CIFAR100
Standalone	96.53	72.53	95.14	62.71	91.97	53.04
FML	30.48	16.84	-	21.96	-	15.21
FedKD	80.20	53.23	77.37	44.27	73.21	37.21
LG-FedAvg	96.30	72.20	94.83	60.95	91.27	45.83
FedClassAvg	94.41	69.16	85.23	56.54	82.61	35.48
FD	96.21	-	-	-	-	-
FedProto	96.51	72.59	95.48	62.69	92.49	53.67
FedSSA	**96.54**	**73.39**	**95.83**	**64.20**	**92.92**	**57.29**

Table 2: The mean test accuracy for *model-heterogeneous* scenarios. N is the total number of clients. C is the fraction of participating clients in each round. '-' denotes failure to converge.

- Knowledge distillation without public data: FD [Jeong *et al.*, 2018], FedProto [Tan *et al.*, 2022b];

- Mutual learning: FML [Shen *et al.*, 2023] and FedKD [Wu *et al.*, 2022];

- Model mixup: LG-FedAvg [Liang *et al.*, 2020] and FedClassAvg [Jang *et al.*, 2022].

Evaluation Metrics. **1) Accuracy**: we measure the *individual test accuracy* (%) of each client's local model and calculate the *average test accuracy* of all clients' local models. **2) Communication Cost**: We trace the number of transmitted parameters when the average model accuracy reaches the target accuracy. **3) Computation Cost**: We track the consumed computation FLOPs when the average model accuracy reaches the target accuracy.

Training Strategy. We tune the optimal FL settings for all methods via grid search. The epochs of local training $E \in \{1, 10\}$ and the batch size of local training $B \in \{64, 128, 256, 512\}$. The optimizer for local training is SGD with learning rate $\eta = 0.01$. We also tune special hyperparameters for the baselines and report the optimal results. We also adjust the hyperparameters μ_0 and T_{stable} to achieve the best-performance FedSSA. To compare FedSSA with the baselines fairly, we set the total number of communication rounds $T \in \{100, 500\}$ to ensure all algorithms converge.

6.2 Comparisons Results

We compare FedSSA with the baselines under both model-homogeneous and model-heterogeneous scenarios with different total numbers of clients N and client participation fraction C to evaluate the robustness of FedSSA. We set up three scenarios: $\{(N = 10, C = 100\%), (N = 50, C = 20\%), (N = 100, C = 10\%)\}$. For ease of comparison across the three settings, $N \times C$ is set to be the same (i.e., 10 clients participate in each round of FL).
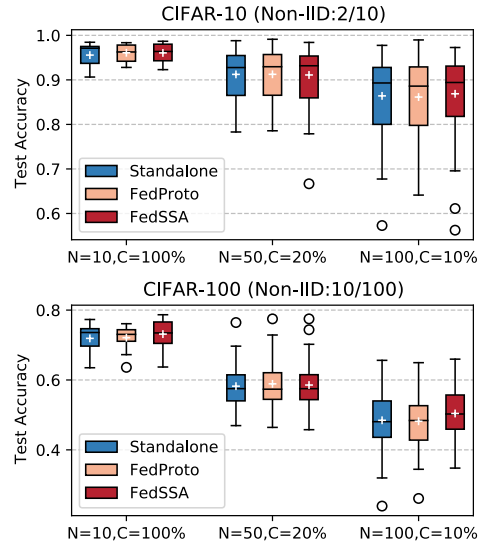

Figure 4: Accuracy distribution for individual clients.

Model-Homogeneous PFL

In this scenario, all clients hold the largest model 'CNN-1' as shown in Table 3. Table 1 shows that FedSSA consistently achieves the highest mean test accuracy under all settings, outperforming the best baseline by 3.61%.

Model-Heterogeneous PFL

We evenly assign 5 models (model IDs: $1 - 5$) with different structures (Table 3, Appendix E) to clients. In practice, the id of the model assigned to client k is determined by (client ID $k\%5$). For FML and FedKD, we regard the 5 structurally different models as large heterogeneous models and consider the smallest 'CNN-5' model as the small homogeneous model.

Mean Accuracy. Table 2 shows that FedSSA consistently achieves the highest mean test accuracy under all settings, outperforming the best baseline by 3.62%. Figure 8 (Appendix E) shows how the mean test accuracy of FedSSA and the two top-performing baselines in each FL setting of Table 2 varies with communication round, it presents that FedSSA converges to higher accuracies faster than others.

Individual Accuracy. Figure 4 shows the accuracy distribution of individual models in FedSSA and the two top-performing baselines in each FL setting shown in Table 2. In Figure 4, '+' denotes the average accuracy of all clients under each algorithm. A small box length bounded by the upper quartile and the lower quartile indicates a more concentrated accuracy distribution across all clients with small variance. We observe that the three algorithms achieve similar mean and variance values in terms of accuracy when $N = \{10, 50\}$, while FedSSA achieves significantly higher mean and lower variance when $N = 100$. This verifies that FedSSA is capable of producing the best personalized heterogeneous models in FL with a large number of potential clients and a low participation rate, which closely reflects practical FL scenarios.

Personalization Analysis. We extract every sample representation from each FL client under FedSSA and FedProto, respectively. Then, we leverage the T-SNE
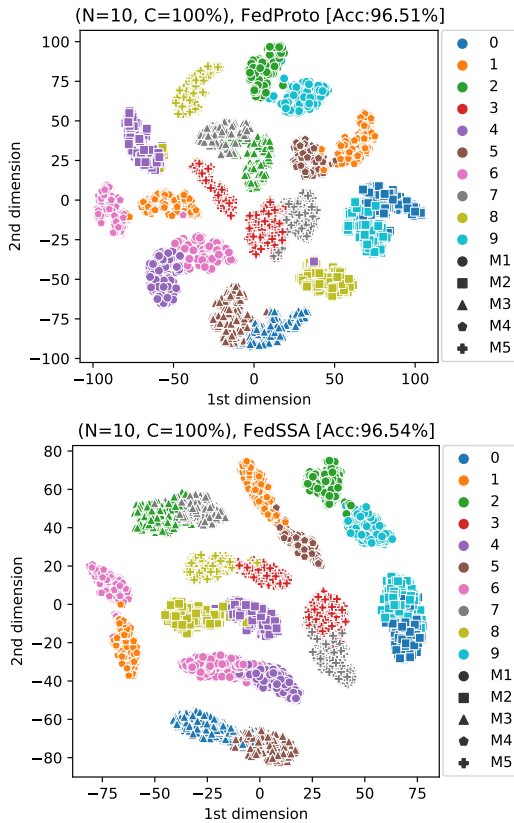
Figure 5: T-SNE representation visualization for `FedProto` and `FedSSA` on the CIFAR-10 (Non-IID: 2/10) dataset. M1-M5 denote 5 heterogeneous models $\{CNN\text{-}1, \ldots, CNN\text{-}5\}$.

[van der Maaten and Hinton, 2008] tool to reduce the dimensionality of the extracted representations from 500 to 2, and visualize the results. Since CIFAR-100 includes 100 classes of samples, we focus on visualizing the results on CIFAR-10 (non-IID: 2/10) in Figure 5. It displays that most clusters in `FedSSA` and `FedProto` consist of representations from a client's two seen classes of samples, indicating that each client's local model has strong personalization capability. Generally, `FedSSA` performs better than `FedProto`.

6.3 Case Studies

Robustness to Non-IIDness (Class)

We study the robustness of `FedSSA` and `FedProto` to non-IIDness controlled by the number of seen classes one client holds under (N=100, C=10%) on CIFAR-10 and CIFAR-100. We vary the number of classes seen by each client as $\{2, 4, \ldots, 10\}$ on CIFAR-10 and $\{10, 30, \ldots, 100\}$ on CIFAR-100. Figure 6 shows that `FedSSA` consistently outperforms `FedProto`, demonstrating its robustness to non-IIDness. As the non-IIDness decreases (the number of classes seen by each client rises), accuracy degrades as more IID local data enhances generalization and reduces personalization.

Robustness to Non-IIDness (Dirichlet)

We also test the robustness of `FedSSA` and `FedProto` to more complex non-IIDness controlled by a Dirichlet(γ)
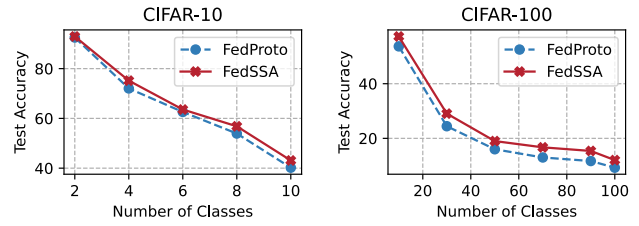

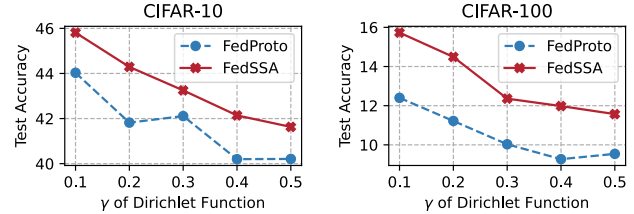Figure 6: Robustness to Non-IIDness (Class).

Figure 7: Robustness to Non-IIDness (Dirichlet).

function under setting (N=100, C=10%) on CIFAR-10 and CIFAR-100. We vary $\gamma = \{0.1, \ldots, 0.5\}$ on both datasets. Figure 7 presents that `FedSSA` also outperforms `FedProto` across all non-IIDnesses [4].

7 Conclusions

In this paper, we propose a novel personalized heterogeneous federated learning framework named `FedSSA` to enhance the performance and efficiency of model-heterogeneous personalized federated learning. It consists of two core modules: 1) local-to-global knowledge transfer by semantic similarity-based header parameter aggregation and 2) global-to-local knowledge transfer by adaptive parameter stabilization-based header parameter fusion, both of them enhance the personalization of each client's heterogeneous local model and stabilize classification boundaries. Theoretical analysis shows that `FedSSA` could converge over wall-to-wall time. Extensive experiments demonstrate that `FedSSA` achieves the best classification accuracy while incurring the lowest communication and computational costs.

Acknowledgments

---

[4]More experiment results (comparisons of communication and computation, robustness to client participant rate, ablation studies) are detailed in Appendices (https://arxiv.org/abs/2312.09006).

References

[Ahn *et al.*, 2019] Jin-Hyun Ahn, Osvaldo Simeone, and Joonhyuk Kang. Wireless federated distillation for distributed edge learning with heterogeneous data. In *Proc. PIMRC*, pages 1–6, Istanbul, Turkey, 2019. IEEE.

[Ahn *et al.*, 2020] Jin-Hyun Ahn, Osvaldo Simeone, and Joonhyuk Kang. Cooperative learning VIA federated distillation OVER fading channels. In *Proc. ICASSP*, pages 8856–8860, Barcelona, Spain, 2020. IEEE.

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

[Chang *et al.*, 2021] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. In *Proc. NeurIPS Workshop*, virtual, 2021. .

[Chen *et al.*, 2021] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. Fedmatch: Federated learning over heterogeneous question answering data. In *Proc. CIKM*, pages 181–190, virtual, 2021. ACM.

[Cheng *et al.*, 2021] Sijie Cheng, Jingwen Wu, Yanghua Xiao, and Yang Liu. Fedgems: Federated learning of larger server models via selective knowledge fusion. *CoRR*, abs/2110.11027, 2021.

[Cho *et al.*, 2022] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *Proc. IJCAI*, pages 2881–2887, virtual, 2022. ijcai.org.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proc. ICML*, volume 139, pages 2089–2099, virtual, 2021. PMLR.

[He *et al.*, 2020] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. In *Proc. NeurIPS*, virtual, 2020. .

[Huang *et al.*, 2022a] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *Proc. CVPR*, pages 10133–10143, virtual, 2022. IEEE.

[Huang *et al.*, 2022b] Wenke Huang, Mang Ye, Bo Du, and Xiang Gao. Few-shot model agnostic federated learning. In *Proc. MM*, pages 7309–7316, Lisboa, Portugal, 2022. ACM.

[Itahara *et al.*, 2023] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Trans. Mob. Comput.*, 22(1):191–205, 2023.

[Jang *et al.*, 2022] Jaehee Jang, Heonseok Ha, Dahuin Jung, and Sungroh Yoon. Fedclassavg: Local representation learning for personalized federated learning on heterogeneous neural networks. In *Proc. ICPP*, pages 76:1–76:10, virtual, 2022. ACM.

[Jeong *et al.*, 2018] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. In *Proc. NeurIPS Workshop on Machine Learning on the Phone and other Consumer Devices*, virtual, 2018. .

[Jiang *et al.*, 2022] Yuang Jiang, Shiqiang Wang, Víctor Valls, Bong Jun Ko, Wei-Han Lee, Kin K. Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *TNNLS*, 2022.

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

[Krizhevsky, 2009] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Toronto, ON, Canada, , 2009.

[Li and Wang, 2019] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. In *Proc. NeurIPS Workshop*, virtual, 2019. .

[Li *et al.*, 2021] Qinbin Li, Bingsheng He, and Dawn Song. Practical one-shot federated learning for cross-silo setting. In *Proc. IJCAI*, pages 1484–1490, virtual, 2021. ijcai.org.

[Li *et al.*, 2023] Xiang Li, Ge Wu, Lingfeng Yang, Wenhai Wang, Renjie Song, and Jian Yang. A survey of historical learning: Learning models with learning history. *CoRR*, abs/2303.12992, 2023.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Ziyin Liu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 1(1), 2020.

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

[Liu *et al.*, 2022] Chang Liu, Yuwen Yang, Xun Cai, Yue Ding, and Hongtao Lu. Completely heterogeneous federated learning. *CoRR*, abs/2210.15865, 2022.

[Liu *et al.*, 2024] Rui Liu, Pengwei Xing, Zichao Deng, Anran Li, Cuntai Guan, and Han Yu. Federated graph neural networks: Overview, techniques, and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2024.

[Makhija *et al.*, 2022] Disha Makhija, Xing Han, Nhat Ho, and Joydeep Ghosh. Architecture agnostic federated learning for neural networks. In *Proc. ICML*, volume 162, pages 14860–14870, virtual, 2022. PMLR.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS, Fort Lauderdale, FL, USA*, volume 54, pages 1273–1282. PMLR, 2017.

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

[Pillutla *et al.*, 2022] Krishna Pillutla, Kshitiz Malik, Abdelrahman Mohamed, Michael G. Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *Proc. ICML*, volume 162, pages 17716–17758, virtual, 2022. PMLR.

[Sattler *et al.*, 2021] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Trans. Neural Networks Learn. Syst.*, 1(1):1–13, 2021.

[Sattler *et al.*, 2022] Felix Sattler, Arturo Marbán, Roman Rischke, and Wojciech Samek. CFD: communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Trans. Netw. Sci. Eng.*, 9(4):2025–2038, 2022.

[Shamsian *et al.*, 2021] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *Proc. ICML*, volume 139, pages 9489–9502, virtual, 2021. PMLR.

[Shen *et al.*, 2023] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Zheqi Lv, Kun Kuang, Chao Wu, and Fei Wu. Federated mutual learning: a collaborative machine learning method for heterogeneous data, models, and objectives. *Frontiers Inf. Technol. Electron. Eng.*, 24(10):1390–1402, 2023.

[Tan *et al.*, 2022a] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Trans. Neural Networks Learn. Syst.*, pages 1–17, 2022.

[Tan *et al.*, 2022b] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proc. AAAI*, pages 8432–8440, virtual, 2022. AAAI Press.

[van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

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

[Yi *et al.*, 2023a] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. Fedgh: Heterogeneous federated learning with generalized global header. In *Proc. MM, Ottawa, ON, Canada*, pages 8686–8696. ACM, 2023.

[Yi *et al.*, 2023b] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. Fedlora: Model-heterogeneous personalized federated learning with lora tuning. *CoRR*, abs/2310.13283, 2023.

[Yi *et al.*, 2023c] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. pfedes: Model heterogeneous personalized federated learning with feature extractor sharing. *CoRR*, abs/2311.06879, 2023.

[Yi *et al.*, 2024a] Liping Yi, Xiaorong Shi, Nan Wang, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. pfedkt: Personalized federated learning with dual knowledge transfer. *Knowledge-Based Systems*, 292:111633, 2024.

[Yi *et al.*, 2024b] Liping Yi, Xiaorong Shi, Nan Wang, Jinsong Zhang, Gang Wang, and Xiaoguang Liu. Fedpe: Adaptive model pruning-expanding for federated learning on mobile devices. *IEEE Transactions on Mobile Computing*, pages 1–18, 2024.

[Yi *et al.*, 2024c] Liping Yi, Han Yu, Chao Ren, Heng Zhang, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedmoe: Data-level personalization with mixture of experts for model-heterogeneous personalized federated learning. *CoRR*, abs/2402.01350, 2024.

[Yu *et al.*, 2011] Han Yu, Siyuan Liu, Alex C. Kot, Chunyan Miao, and Cyril Leung. Dynamic witness selection for trustworthy distributed cooperative sensing in cognitive radio networks. In *2011 IEEE 13th International Conference on Communication Technology*, pages 1–6, 2011.

[Yu *et al.*, 2022] Sixing Yu, Wei Qian, and Ali Jannesari. Resource-aware federated learning using knowledge extraction and multi-model fusion. *CoRR*, abs/2208.07978, 2022.

[Zhang *et al.*, 2021] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *Proc. NeurIPS*, pages 10092–10104, virtual, 2021.

[Zhang *et al.*, 2023] Yuxin Zhang, Jindong Wang, Yiqiang Chen, Han Yu, and Tao Qin. Adaptive memory networks with self-supervised learning for unsupervised anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12068–12080, 2023.

[Zhu and others, 2021] Zhuangdi Zhu et al. Data-free knowledge distillation for heterogeneous federated learning. In *Proc. ICML*, volume 139, pages 12878–12889, virtual, 2021. PMLR.

[Zhu *et al.*, 2021] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[Zhu *et al.*, 2022] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Fedzkt: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *Proc. ICDCS*, pages 928–938, virtual, 2022. IEEE.