This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Closing the Generalization Gap of Cross-silo Federated Medical Image Segmentation

An Xu* Univ. of Pittsburgh
Wenqi Li NVIDIA
Pengfei Guo Johns Hopkins University
Dong Yang NVIDIA
Holger Roth NVIDIA
Ali Hatamizadeh NVIDIA
Can Zhao NVIDIA
Daguang Xu NVIDIA
Heng Huang Univ. of Pittsburgh
Ziyue Xu NVIDIA

Abstract

Cross-silo federated learning (FL) has attracted much attention in medical imaging analysis with deep learning in recent years as it can resolve the critical issues of insufficient data, data privacy, and training efficiency. However, there can be a generalization gap between the model trained from FL and the one from centralized training. This important issue comes from the non-iid data distribution of the local data in the participating clients and is well-known as client drift. In this work, we propose a novel training framework FedSM to avoid the client drift issue and successfully close the generalization gap compared with the centralized training for medical image segmentation tasks for the first time. We also propose a novel personalized FL objective formulation and a new method SoftPull to solve it in our proposed framework FedSM. We conduct rigorous theoretical analysis to guarantee its convergence for optimizing the non-convex smooth objective function. Real-world medical image segmentation experiments using deep FL validate the motivations and effectiveness of our proposed method.

1. Introduction

Deep learning models have shown success in computer vision tasks in recent years [19, 41, 42]. However, training deep models that generalize well on unseen test data may require massive training data. Unfortunately, we are usually faced with **insufficient data** in a single medical institution for the medical image segmentation task due to the expensive procedure of collecting enough patients' data with experts' labeling.

A straightforward solution to address the insufficient data issue is gathering data from all the available medical institutions, while the amount of data owned by any single institution may be insufficient to train a well-performing deep model. However, this approach will raise the concern for **data privacy**. On one hand, collecting medical data is expensive as mentioned above, and those data have become a valuable asset at a medical institution. Institutions with more data may be more reluctant to contribute their data. In addition, medical institutions bear the obligation to keep the data collected from patients secure. Gathering data may expose patients to the risk of data leakage.

Of course, we can leverage the existing vanilla distributed training method [31, 49, 50] to keep the institution's data local and share only the gradient with a central server. But the training of deep model requires many iterations to converge, leading to unacceptable **communication complexity** for vanilla distributed training. It is not secure neither as recent works [14, 52, 54, 55] have shown that pixel-level images can be recovered from the leaked gradient.

Recently, federated learning (FL) [13, 16, 29, 48] have been proposed to tackle all the above issues (insufficient data, data privacy, training efficiency). In medical applications, we are most interested in the cross-silo federated learning where we have a limited number of participating clients compared with cross-device federated learning (e.g., mobile devices) [17, 26, 36]. Specifically, in each training round of FedAvg [38], the *de facto* algorithm for FL, each client will perform local training with the global model received from a central server for multiple iterations. Then the server gathers all the local models from each client and averages them as the new global model. Nevertheless, for FedAvg and its variants, a non-negligible issue called "client drift" arises due to non-iid data distribution on different clients. The local models on different clients will gradually diverge from each other during the local training. Client drift can drastically jeopardize the training performance of the global model when the data similarity decreases (more non-iid) [20, 21]. Theoretically, it leads to a convergence rate more sensitive to the number of local

*Work done during an internship at NVIDIA. A.X. and H.H. were partially supported by NSF IIS 1845666, 1852606, 1838627, 1837956, 1956002, IIA 2040588. Implementation of this work is available at https://github.com/NVIDIA/NVFlare/examples/FedSM

training steps [53].

Throughout this paper, we refer to centralized training as gathering data from clients and then training the model. Note that centralized training is impractical as it violates data privacy, but offers a performance upper bound for FL algorithms. Despite numerous efforts and previous works, there is still a **generalization gap** between FL and the centralized training. In this paper, unlike any previous works, we propose a novel training framework called Federated Super Model (FedSM) to avoid confronting the difficult client drift issue at all for FL medical image segmentation tasks. In FedSM, instead of finding one global model that fits all clients' data distribution, we propose to produce personalized models to fit different data distributions well and a novel model selector to decide the closest model/data distribution for any test data.

We summarize our contributions as follows.

- We propose a novel training framework FedSM to avoid the client drift issue and close the generalization gap between FL and centralized training for medical segmentation tasks for the first time to the best of our knowledge.
- We propose a novel formulation for personalized FL optimization, and a novel personalized method called SoftPull to solve it in our framework FedSM. A rigorous convergence analysis with common assumptions in FL is given for the proposed method.
- Experiments in real-world FL medical image segmentation tasks validate our motivation and the superiority of our methods over existing FL baselines.

2. Related Works

Here we introduce existing different approaches to improve the model performance in FL with representative methods. First, the FL optimization problem is usually defined as $\min_w \frac{1}{K}\sum_{k=1}^{K} p_k L_{\mathcal{D}_k}(w)$, where the coefficient $p_k = \frac{n_k}{n}$, n_k is the number of client k's data, and the total number of data $n = \sum_{k=1}^{K} n_k$. $L_{\mathcal{D}_k}$ is the objective at client k with its local data $\mathcal{D}_k$, and w is the model weights.

FedAvg. In FedAvg, clients will receive the starting model w_r from the server at training round r. Each client k performs E epochs of local training to update the local model to $w_{r+1}^{(k)}$ with the popular momentum SGD or Adam [28] optimizer depending on the application needs. Then the server gathers and averages the local models to $w_{r+1} = \frac{1}{K}\sum_{k=1}^{K} p_k w_{r+1}^{(k)}$.

Restrict Local Training. To discourage the local models from diverging due to non-iid data distribution, FedProx [33] proposes to add a proximal loss term $\|w_{r+1}^{(k)} - w_r\|_2^2$ to the objective function for client k. It implies that the local training will encourage $w_{r+1}^{(k)}$ to stay close to the starting point w_r, such that $\{w_{r+1}^{(k)}\}_{k\in\{1,2,\cdots,K\}}$ will be close to each other to alleviate the client drift issue.

Correct Client Drift. Motivated by variance reduction techniques in optimization such as SVRG [25], SAGA [8], inter-client variance reduction techniques [5,27,35] are proposed for FL by correcting the local training with the predicted local and global updating direction. These methods are usually tested with convex or simple non-convex models/objectives. For the practical training of complicated deep models, [9] shows that variance reduction techniques fail to perform well in that correcting the stochastic gradient with variance reduction usually does not hold in deep learning due to common augmentation tricks such as batch normalization [22] and dropout [44], etc.

Personalization. Personalized models are usually a fine-tuned version of the global model to better fit the local data distribution of a specific client. We can fine-tune the global model [47] on a client's local data like the local training, or following MAML-based personalized methods [11, 24, 45]. However, an intrinsic drawback of the personalized models is that they generalize poorly on other sites' data and unseen data. In this work, we focus on finding a model that generalizes as well as centralized training for all clients.

Other Topics. There are also many other emerging and interesting topics in FL, such as heterogeneous optimization [33, 46], fairness and robustness [32, 34, 39], clustered federated learning [15], etc. These topics are not directly related to our work but can be valuable for potential future extension. A recent work FedDG [36] requires sharing partial information of the data, therefore it breaks the data privacy constraint to some extent. In this work, we share only the model update information for maximal data privacy.

3. Methodology

In this section, we present our motivation and the proposed method that can close the generalization gap for FL medical image segmentation tasks in detail.

Motivation. In traditional FL, the goal is to collaboratively train one global model that generalizes well on all clients' joint data distribution. The client drift issue comes from the fact that we only have access to clients' local data distribution during the local training. It is hard to train a global model generalizing as well as centralized training due to this issue despite numerous existing works. In this work, however, we show that it is possible to get rid of the client drift issue. Specifically, we propose that

- for the test data, we search for the closest (i.e., the most similar) local data distribution from all clients (Section 3.1).
- we find a model with the best generalization performance on this selected local data distribution, and use it for the inference of the test data (Section 3.2).

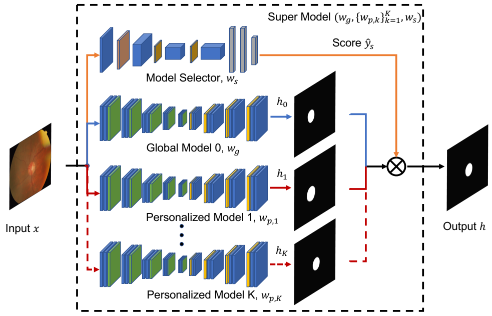
Super Model $(w_g, \{w_{p,k}\}_{k=1}^K, w_s)$
Score $\hat{y}_s$
Model Selector, w_s
h_0
Global Model 0, w_g
h_1
Personalized Model 1, $w_{p,1}$
h_K
Personalized Model K, $w_{p,K}$
Input x
Output h

Figure 1. The proposed FedSM framework with “super model”.

3.1. New Framework: FedSM

The first motivation above motivates us to design a new and general FL framework FedSM, where we train a Federated “Super Model” consisting of the global model, personalized models, and a model selector. These components are illustrated in Figure 1 and we elaborate them as follows.

Global model w_g: the global model trained by FedAvg. It generalizes better than personalized models on the joint data distribution of all clients, but there is still a gap compared with centralized training. Suppose the model function is f and we denote its output as $h_0 = f(w_g, x)$ for data x.

Personalized models $w_{p,k}$: the personalized models trained by any personalization FL training method. A personalized model usually generalizes better on local data than the global model. We denote its output as $h_k = f(w_{p,k}, x)$, where $k \in \{1, 2, \cdots, K\}$.

Model selector w_s: its goal is to determine the match between the unseen data input x and each of the global/personalized models for inference. Specifically, it outputs a normalized prediction score vector $\widehat{y}_s$. The final output h is determined by $\widehat{y}_s$ and $[h_0, h_1, \cdots, h_K]$. Suppose the candidate model set $\Omega \subseteq \{0, 1, 2, \cdots, K\}$, then $\sum_{k\in\Omega} \widehat{y}_{s,k} = 1$ and $h = \sum_{k\in\Omega} \widehat{y}_{s,k} h_k$. We discuss the potential training methods as follows.

3.1.1 Ensemble

Suppose we already have the trained global model and personalized models. Given the FedSM framework as shown in Figure 1, a straightforward approach is to ensemble the outputs $[h_0, h_1, \cdots, h_K]$ from all models as the final output $h = \sum_{k=0}^K \widehat{y}_{s,k} h_k$. Let the ground truth of data x be y and the loss function be L. Then, we compute the loss $L(h, y)$ and update the model selector w_s via FedAvg.

However, in practice we find it **hard to train** the model selector in this way in FL. The final performance can be even inferior to the global model. Let the desired value $y_s = \min_{\widehat{y}_s} L(\sum_{k=0}^K \widehat{y}_{s,k} h_k, y)$. We found that it was caused by the difficulty to train $\widehat{y}_s$ to the desired value y_s by $\min_{w_s} L(\sum_{k=0}^K \widehat{y}_{s,k} h_k, y)$ as w_s is the model weights to optimize. For each data input x, we may need many training steps to $\min_{w_s} L(\sum_{k=0}^K \widehat{y}_{s,k} h_k, y)$ such that $\widehat{y}_s$ will be close to y_s. However, it is unacceptable due to the large amount of computation cost.

Another issue of this approach is that we cannot start training the model selector until the training of the global model and personalized models finishes, which incurs **extra communication rounds** for FL.

3.1.2 FedSM-extra

To tackle the **training difficulty** in ensemble, here we propose to compute

$$y_s = one\_hot(\arg\min_k \{L(h, h_k)\}_{k=0}^K) \,, \tag{1}$$

where “one_hot” denotes one hot encoding. Then we compute the cross entropy loss $L_s(\widehat{y}_s, y_s)$ to update the model selector. In this way, the model selector is more clear about the desired value y_s. Thus it will be easier to train. We refer to this approach as FedSM-extra as it still needs extra communication rounds like the ensemble approach.

3.1.3 FedSM

To address the issue of **extra training rounds**, the model selector needs to be trained together with the global model and personalized models. Nevertheless, from Eq. (1) we can see that the desired y_s depends on the output of the trained global model and personalized models. Therefore, we need to decouple their dependency. As a further simplification, suppose the training data x comes from the client $k \in \{1, 2, \cdots, K\}$, here we propose

$$y_s = one\_hot(k) \,. \tag{2}$$

Intuitively, the personalized model k tends to generalize better on client k’s own local data. It is safe to set y_s as the corresponding client index. Though theoretically, it may degrade the performance of Eq. (1), it is more practical due to no extra training rounds. We refer to this approach as FedSM which addresses all the issues raised by the ensemble.

3.2. New Personalization: SoftPull

In this section, we present a new personalized FL optimization formulation and a method, SoftPull, to solve it and produce personalized models for FedSM. We first present existing interpolation methods to tackle the insufficient local data issue.

Let the global dataset be $\mathcal{D}$. To tackle the insufficient local data issue, [37] proposes dataset interpolation for each client as $\min_{w_{p,k}} \lambda L_{\mathcal{D}_k}(w_{p,k}) + (1-\lambda) L_{\mathcal{D}}(w_{p,k})$, where coefficient $\lambda \in [0,1]$. As client $k \in \{1, 2, \cdots, K\}$, it leads to K optimization problems and is inefficient to solve. Besides, it is hard to acquire the information of the global dataset D during the local training. [37] also proposes model interpolation $\min_{w_g, w_{p,k}, \lambda} \sum_{k=1}^{K} L_{\mathcal{D}_k}(\lambda w_{p,k} + (1-\lambda) w_g)$. To efficiently solve the model interpolation problem, APFL [10] proposes

$$w_g^* = \arg\min_{w_g} L_{\mathcal{D}}(w_g)\,, \tag{3}$$

$$w_{p,k}^* = \arg\min L_{\mathcal{D}_k}(\lambda w_{p,k} + (1-\lambda) w_g^*)\,, \tag{4}$$

$$w_{p.k} \leftarrow \lambda w_{p,k}^* + (1-\lambda) w_g^*\,. \tag{5}$$

Motivation. We observe that model interpolation tries to find an appropriate combination between the FL global and local models. When the local data distribution is not similar to the global data distribution at all, we expect $\lambda \to 1$. When they are similar, we expect $\lambda \to \frac{1}{K}$ to leverage the global data information to improve the local generalization as the local dataset is small. Nevertheless, the formulation of APFL has two potential drawbacks:

- The involved global model w_g^* may not generalize well on $\mathcal{D}$ and $\mathcal{D}_k$, but will affect the FL training.
- What objective function it is exactly optimizing is not clear.

In our problem formulation, we first suppose w_k^* is the local optimum of client k:

$$w_k^* = \arg\min_{w} L_{\mathcal{D}_k}(w)\,. \tag{6}$$

However, local optimum w_k^* may not generalizes well due to lack of local training data. Instead of interpolating the global and local optimum, we propose that the desired *personalized optimum $w_{p,k}^*$ is an interpolation between the local optimum of client k and other clients' personalized optima*:

$$w_{p,k}^* = \lambda w_k^* + (1-\lambda) \frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{p,k'}^*\,. \tag{7}$$

The new interpolation avoids the global model and guarantees that the interpolated model is the optimum to some explicit objective function, as opposed to APFL. In fact, the personalized optimum $w_{p,k}^*$ is also an interpolation between the local optimum of client k and other clients' local optimum because Eq. (7) is identical to

$$w_{p,k}^* = \lambda w_k^* + (1-\lambda) \frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{k'}^*\,. \tag{8}$$

However, Eq. (7) is better to help us to find what objective function we are optimizing as we can turn it to

$$w_k^* = \frac{1}{\lambda} w_{p,k}^* - \frac{1-\lambda}{\lambda} \frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{p,k'}^*\,. \tag{9}$$

Compare it with Eq. (6) and we immediately have $\{w_{p,k}^*\}_{k=1}^{K}$ as the solution to the optimization problem

$$\min_{\{w_{p,k}\}} \sum_{k=1}^{K} L_{\mathcal{D}_k}\Big(\frac{1}{\lambda} w_{p,k} - \frac{1-\lambda}{\lambda} \frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{p,k'}\Big)\,. \tag{10}$$

To solve the proposed new personalized FL optimization problem Eq. (10), we propose a new method, SoftPull ($\lambda \in [\frac{1}{K}, 1]$), with the simplification of substituting w_k^* with the locally trained model in Eq. (7), that is, after each training round at the server,

$$w_{p,k} \leftarrow \lambda w_{p,k} + (1-\lambda) \frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{p,k'}\,. \tag{11}$$

The corresponding algorithm is summarized in Algorithm 1, line 16. When $\lambda = \frac{1}{K}$, it reduces to the "hard" averaging in FedAvg. To analyze the convergence, we start with common assumptions as follows.

Assumption 1 *(Lipschitz Smooth) The loss function $L_{\mathcal{D}_k}$ is L-smooth, that is, $\forall w_1, w_2 \in \mathbb{R}^d$, we have*

$$\|\nabla L_{\mathcal{D}_k}(w_1) - \nabla L_{\mathcal{D}_k}(w_2)\|_2^2 \leq L \|w_1 - w_2\|_2^2\,. \tag{12}$$

Assumption 2 *(Bounded Variance) The stochastic gradient $\nabla L_{\mathcal{D}_k}(w, x)$ has bounded variance $\forall w \in \mathbb{R}^d$:*

$$\mathbb{E}\|\nabla L_{\mathcal{D}_k}(w, x) - \nabla L_{\mathcal{D}_k}(w)\|_2^2 \leq \sigma^2\,. \tag{13}$$

where $\mathbb{E}$ is an expectation over $x \in \mathcal{D}_k$.

Assumption 3 *[40] The gradient $\nabla L_{\mathcal{D}_k}(w)$ has bounded value $\forall w \in \mathbb{R}^d$: $\|\nabla L_{\mathcal{D}_k}(w)\|_2^2 \leq G^2$.*

Theorem 1 *Suppose Assumptions 1, 2, and 3 exist. Let the proposed objective in Eq. (10) be F, superscript (r, m) denote the global iteration, and $\overline{w}$ denote the average, then*

$$\frac{1}{KRM} \sum_{r=0}^{R-1} \sum_{m=0}^{M-1} \sum_{k=1}^{K} \mathbb{E}\|\nabla_{w_{p,k}^{r,m}} F\|_2^2 \tag{14}$$

$$= \mathcal{O}\Big(\frac{1}{\eta RM\lambda^2} + \frac{(1-\lambda)^2}{KRM\eta^2\lambda^2} \sum_{K=1}^{K} \sum_{r=0}^{R-1} \mathbb{E}\|w_{p,k}^{r,M} - \overline{w}_{p,k}^{r,M}\|_2^2$$

$$+ \frac{(1-\lambda)^2}{KRM\lambda^4} \sum_{k=1}^{K} \sum_{r=0}^{R-1} \sum_{m=0}^{M-1} \mathbb{E}\|w_{p,k}^{r,m} - \overline{w}_{p,k}^{r,m}\|_2^2\Big)$$

$$= \mathcal{O}\Big(\frac{1}{\eta RM\lambda^2} + \frac{M\sum_{r=0}^{R-1}(1-\lambda)^2}{R\lambda^2} + \frac{M^2\eta^2\sum_{r=0}^{R-1}(1-\lambda)^2}{R\lambda^4}\Big)\,.$$

If $\eta = \mathcal{O}(\frac{1}{\sqrt{RM}})$ and $M = \mathcal{O}(R^{\frac{1}{3}})$, its convergence rate is $\mathcal{O}(\frac{1}{\sqrt{RM}})$ with a convergence error $\mathcal{O}(\frac{M\sum_{r=0}^{R-1}(1-\lambda)^2}{R\lambda^2})$.

Algorithm 1 FedSM training.

```
Input: local dataset $\mathcal{D}_k$, rounds $R$, number of sites $K$, learning rate $\eta$, $\eta_s$, coefficient $\lambda$, client weight $\frac{n_k}{n}$.
Initialize: global model $w_g$, personalized model $w_{p,k}$, model selector $w_s$, base optimizer OPT$(\cdot)$
for round $r = 1, 2, \cdots, R$ do
    SERVER: send models ($w_g$, $w_{p,k}$, $w_s$) to client $k$.
    for CLIENT $k \in \{1, 2, \cdots, K\}$ in parallel do
        initialize $w_{g,k} \leftarrow w_g$, $w_{s,k} \leftarrow w_s$
        for batch $(x, y) \in \mathcal{D}_k$ do
            $w_{g,k} \leftarrow \text{OPT}(w_{g,k}, \eta, \nabla_{w_{g,k}} L(f(w_{g,k}; x), y))$
            $w_{p,k} \leftarrow \text{OPT}(w_{p,k}, \eta, \nabla_{w_{p,k}} L(f(w_{p,k}; x), y))$
            // $y_s$ from Eq. (2)
            $w_{s,k} \leftarrow \text{OPT}(w_{s,k}, \eta_s, \nabla_{w_{s,k}} L_s(f_s(w_{s,k}; x), y_s))$
        end for
        send ($w_{g,k}$, $w_{p,k}$, $w_{s,k}$) to server
    end for
    SERVER: $w_g, w_s \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{g,k}, \sum_{k=1}^{K} \frac{n_k}{n} w_{s,k}$
    SERVER: $\forall k \in \{1, 2, \cdots, K\}$, $w_{p,k} \leftarrow \lambda w_{p,k} + (1 - \lambda)\frac{1}{K-1} \sum_{k'=1, k' \neq k}^{K} w_{p,k'}$    // SoftPull
end for
Output: model ($w_g$, $\{w_{p,k}\}_{k=1}^{K}$, $w_s$)
```

Algorithm 2 FedSM inference.

```
Input: data $x$, model ($w_g$, $\{w_{p,k}\}_{k=1}^{K}$, $w_s$), threshold $\gamma$
$\widehat{y}_s = f_s(w_s; x)$
if max$(\widehat{y}_s) > \gamma$ then
    $k = \arg\max(\widehat{y}_s) \in \{1, 2, \cdots, K\}$// high confidence
    $\widehat{y} = f(w_{p,k}; x)$
else
    $\widehat{y} = f(w_g; x)$    // low confidence
end if
Output: $\widehat{y}$
```

Remark 1.1 *When the data similarity is low among clients, we should set a larger λ to reduce the effect of $\|w_{p,k}^{r,m} - \overline{w}_{p,k}^{r,m}\|_2^2$ and ensure the convergence rate. It is intuitively valid as the client has less to learn from other clients.*

Remark 1.2 *$\lambda \downarrow$ and the convergence error $\uparrow$, but it does not mean worse generalization because we do not want to overfit local data. We will empirically tune and validate it.*

The proof can be found in Appendix C.

3.3. All Together

We summarize the proposed SoftPull method to train personalized models and the FedSM framework consisting of the model selector, global model, and personalized models in Algorithm 1. Compared with FedAvg, the communication cost of each training round is $2w_g + w_s$ for FedSM.

Client	1	2	3	4	5	6	Global
Train	50	98	47	230	80	400	905
Val	25	49	24	115	40	200	453
Test	26	48	23	115	39	200	451

Table 1. Retinal Dataset: number of data (2D image) in each client. The data sources from client 1 to 6 are Drishti-GS1 [43], RIGA [6] BinRushed, RIGA Magrabia, RIGA MESSIDOR, RIM-ONE [12], and REFUGE [4] respectively. Global refers to the data from all clients.

Client	1	2	3	4	5	6	Global
Train	153	404	464	361	609	1179	3170
Val	77	215	219	162	289	582	1544
Test	61	245	198	150	329	532	1515

Table 2. Prostate Dataset: number of data (2D slices) in each client. The data sources from client 1 to 6 are I2CVB [30], MSD [7], NCI_ISBI_3T, NCI_ISBI_DX [1], Promise12 [2], and ProstateX [3] respectively. Global refers to the data from all clients.

We note that some methods such as Scaffold [27] have a cost of $2w_g$. After the training, the server sends the super model (w_g, $\{w_{p,k}\}_{k=1}^{K}$, w_s) to each client for inference, which incurs only a one-time communication cost.

For the FedSM inference in Algorithm 2, we propose a heuristic technique that the model selector selects the global model when its confidence is low, because we do not have label 0 in Eq. (2) (the global model) during training. Intuitively, if the test data is not similar to any local data distribution, the global model should be a better choice for its inference, in that it covers the joint data distribution while the personalized model covers only one local data distribution. It also guarantees that FedSM is at least not worse than the global model from FedAvg with an appropriate threshold γ.

For FedSM-extra, both the training and inference algorithms are the same except for the determination of y_s, the extra training rounds, and no need for the threshold γ. More details are available in Appendix B.

4. Experiments

We validate our proposed method on three real-world FL medical image segmentation tasks: retinal disc & cup from 2D fundus images, and prostate segmentation from 3D MR images. The global and personalized model architecture is 2D U-Net [41], while the model selector architecture is VGG-11 [42]. We randomly split the data to train/validation/test with a ratio of 0.5/0.25/0.25. The image data are resized to 256×256. The local training epoch is 1 and the total training rounds is 150. Most methods converge in 100 rounds. But for FedSM-extra, we train the global and personalized models for 100 rounds and the model selector

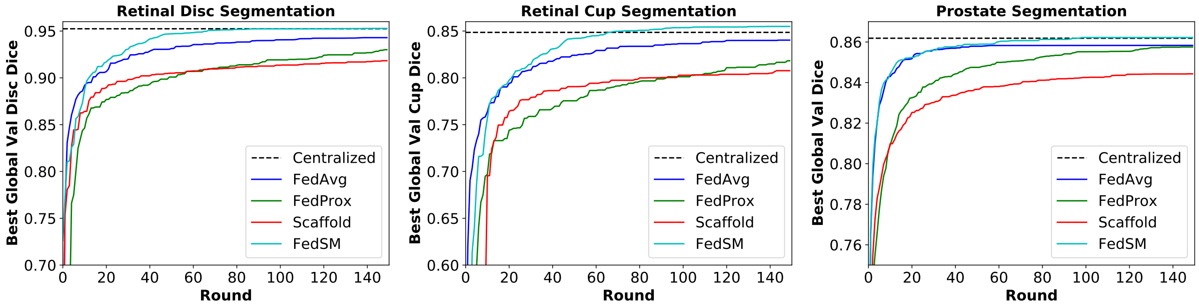

Figure 2. Training curves comparison. The curves are non-decreasing because we record the best result during training.

Method	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client Avg Dice	Global Dice
Centralized	0.9161	0.8760	0.8758	0.9022	0.8510	0.9179	0.8898	0.9014
Client 1 Local	0.8835	0.3331	0.7345	0.4933	0.3408	0.7015	0.5811	0.5902
Client 2 Local	0.2346	0.8620	0.0886	0.7751	0.1791	0.4106	0.4250	0.5050
Client 3 Local	0.8337	0.3402	0.8766	0.6010	0.3644	0.7794	0.6326	0.6594
Client 4 Local	0.5108	0.8574	0.3457	0.9008	0.2361	0.6822	0.5888	0.6910
Client 5 Local	0.5241	0.1584	0.3953	0.2039	0.8223	0.6222	0.4544	0.4662
Client 6 Local	0.7908	0.6649	0.7325	0.7681	0.3742	0.9150	0.7076	0.7877
FedAvg	0.8847	0.8679	0.8667	0.9015	0.7877	0.9172	0.8710	0.8923
FedProx	0.8635	0.8522	0.8547	0.8952	0.6852	0.9095	0.8434	0.8749
Scaffold	0.8380	0.8513	0.8215	0.8935	0.5671	0.9130	0.8141	0.8625
FedSM	**0.9132**	**0.8769**	**0.8865**	**0.9041**	**0.8483**	**0.9195**	**0.8914**	**0.9028**
FedSM-extra	**0.9134**	**0.8763**	**0.8841**	**0.9038**	**0.8483**	**0.9172**	**0.8905**	**0.9007**

Table 3. (low data similarity) Test Dice coefficient comparison of retinal segmentation. "Client k Local" refers to local training on client k. The first row refers to the performance on client 1∼6's test data, their average, and the performance on all clients' test data. We report the average of disc and cup Dice coefficients here. We bold the best FL numbers. See Appendix D for their separate numbers and the visual comparison of segmentation.

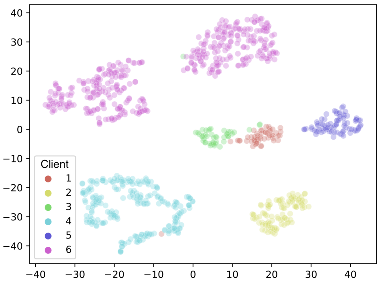

Figure 3. TSNE map of the features extracted form the model selector on retinal segmentation task.

for an extra 50 rounds. The loss function is Dice loss and the test metric is Dice coefficient. The base optimizer is Adam with $\beta = (0.9, 0.999)$. We tune the best learning rate for all methods and the threshold γ for FedSM. For prostate segmentation, in particular, the image data are 3D but we take the 2D slices and perform 2D segmentation. Each experiment repeatedly runs 3 times and we report the mean value.

The dataset information is summarized in Table 1 and 2. Overall, the retinal dataset features lower data similarity among clients (stronger non-iid). The images may differ in position, color, brightness, background ratio, etc. While the prostate dataset has a higher data similarity as the images mostly differ in brightness (see Appendix A).

We compare FedSM and FedSM-extra with baselines (1) Centralized: centralized training, which is the upper bound but prohibited in FL, (2) Local: local training on one client, (3) FedAvg [38], the *de facto* FL method, (4) FedProx [33], and (5) Scaffold [27].

4.1. General Results

We compare the training curves of different methods in Figure 2. The centralized training upper bound is plotted as a horizontal dash line. We can see that the proposed FedSM is the only FL method to close the validation gap to centralized training. FedSM is even better than centralized training on the retinal cup segmentation task, due to the proposed

Method	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client Avg Dice	Global Dice
Centralized	0.9018	0.8583	0.8702	0.8844	0.8800	0.8474	0.8737	0.8651
Client 1 Local	0.8582	0.3886	0.4476	0.2849	0.3830	0.4697	0.4720	0.4336
Client 2 Local	0.7166	0.7669	0.8317	0.7341	0.6156	0.7754	0.7401	0.7403
Client 3 Local	0.6470	0.8541	0.8549	0.6735	0.6591	0.7519	0.7401	0.7496
Client 4 Local	0.4515	0.6566	0.6700	0.8518	0.4558	0.6267	0.6187	0.6148
Client 5 Local	0.8198	0.7751	0.8469	0.8029	0.8038	0.7928	0.8069	0.8016
Client 6 Local	0.8555	0.7965	0.8260	0.7206	0.6478	0.8466	0.7822	0.7809
FedAvg	0.8775	0.8575	0.8700	0.8802	0.8717	0.8532	0.8684	0.8638
FedProx	**0.8948**	0.8511	0.8722	0.8803	0.8668	0.8513	0.8694	0.8621
Scaffold	0.8500	0.8440	0.8570	0.8423	0.8431	0.8412	0.8463	0.8446
FedSM	**0.8946**	**0.8596**	**0.8786**	**0.8898**	**0.8817**	**0.8535**	**0.8763**	**0.8692**
FedSM-extra	0.8886	**0.8584**	**0.8766**	**0.8880**	**0.8760**	**0.8542**	**0.8736**	**0.8673**

Table 4. (high data similarity) Test Dice coefficient comparison of prostate segmentation. We bold the best FL numbers. See Appendix D for the visual comparison.

Unseen Client k	Threshold γ	GM	PM1	PM2	PM3	PM4	PM5	PM6	Dice	Best γ, Dice
Client $k = 6$	0	0	0.02	0	0.35	0	0.63	N/A	0.8587	1, 0.8906
Client $k = 5$	0	0	0.31	0.03	0	0.61	N/A	0.05	0.4015	0.9, 0.4304
Client $k = 4$	0	0	0	1.00	0	N/A	0	0	0.8869	<0.95, 0.8870
Client $k = 3$	0	0	0	0.57	N/A	0	0	0.43	0.8441	<0.9, 0.8446
Client $k = 2$	0	0	0	N/A	0	0.92	0.08	0	0.8409	<1, 0.8409
Client $k = 1$	0	0	N/A	0	1.00	0	0	0	0.8839	<0.99, 0.8839

Table 5. (retinal segmentation, Dice = average of disc and cup Dice coefficients) Model selection frequency from the model selector when FL train with clients $\{1, 2, \cdots, 6\}/\{k\}$ and test on the **unseen** client $k \in \{1, 2, \cdots, 6\}$. From left to right, GM denotes the global model and PM denotes the personalized model $\{1, 2, \cdots, 6\}/\{k\}$. The model selection frequency with the best γ, and the more detailed Dice results can be found in Appendix D. Note GM is never selected as the Threshold γ is intentionally set to 0.

SoftPull personalization method. Note that we can not show the training curve of FedSM-extra as its model selector has to be trained in the extra training rounds.

We summarize the testing numbers in Table 3 and 4. For retinal segmentation, FedSM slightly improves centralized training regarding the client average Dice and global Dice by 0.2% and 0.1% respectively, while FedAvg shows a decrease of 1.9% and 0.9%. The FedSM-extra shows the same performance as FedSM, validating the proposed simplification from Eq. (1) to Eq. (2). For prostate segmentation, similar patterns can be observed. But the gap becomes smaller due to higher data similarity among clients.

For retinal segmentation, FedSM outperforms centralized training for client 3 and matches centralized training for the other clients. However, FedAvg is inferior to centralized training for clients 1, 2, 3, and 5 where the local dataset size is smaller. What's more, FedAvg shows similar test Dice performance to local training for clients 1 and 2, and is even inferior to local training for clients 3 and 5. Therefore, those clients do not benefit from FL via FedAvg, and may not be willing to join the FL system.

We also observe that local training does not generalize well on other clients' data, which is critical as it will perform poorly for patients from other clients (medical institutions). Centralized training improves the local training on the local dataset, especially for clients with insufficient data.

4.2. Validate Motivation

Validate FedSM. Recall that our first motivation is to find the closest local data distribution for the test data. In FedSM, we first plot the TSNE map of the features extracted from the model selector in Figure 3. To validate that the model selector can fulfill our motivation, we sequentially choose client $k \in \{1, 2, \cdots, 6\}$ as the unseen client to test and FL train the model with clients $\{1, 2, \cdots, 6\}/\{k\}$. We set the threshold $\gamma = 0$ to let the model selector select from the personalized models. We summarize the frequency in Table 5. We can see that the model selector tends to select the personalized models of clients 3 and 5 for client 6, which also matches Figure 3 and the local training results in Table 3 that clients 3 and 5 are more similar to client 6. Similar patterns can be observed for the other clients. Therefore, the model selector indeed fulfills our motivation. Note that to validate the model selector, we cannot let the unseen client k join the FL system. Because in that case, the model selector tends to select its own personalized model.

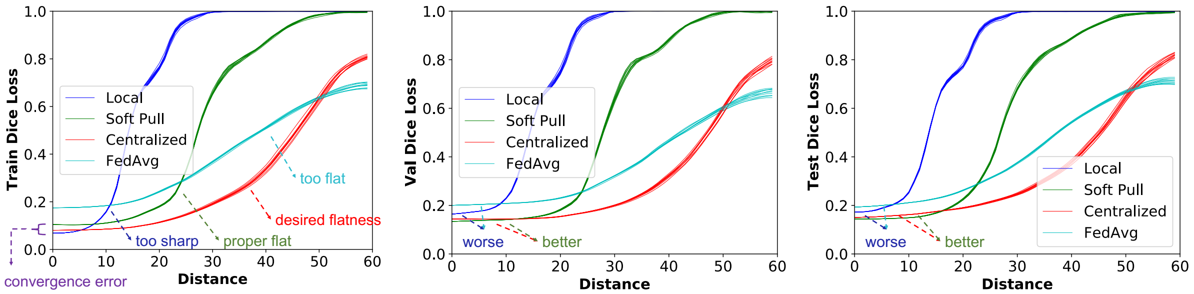

Figure 4. The 1D loss surface near the models trained by different methods on Client 5's data in retinal segmentation.

Method	Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client Avg Dice	Global Dice
FT [47]	0.9087	0.8703	0.8877	0.9003	0.8409	0.9151	0.8875	0.8984
APFL [10]	0.9083	0.8640	0.8794	0.8969	0.8416	0.9152	0.8842	0.8966
Per-FedAvg [11]	0.9051	0.8559	0.8708	0.8954	0.8031	0.9119	0.8737	0.8900
Per-FedMe [45]	0.9084	0.8646	0.8822	0.8980	0.8211	0.9162	0.8818	0.8957
SoftPull	**0.9132**	**0.8769**	**0.8865**	**0.9041**	**0.8483**	**0.9195**	**0.8914**	**0.9028**

Table 6. FedSM with different personalization method in retinal segmentation. Dice = average of disc and cup Dice coefficients.

λ	0.1	0.3	0.5	**0.7**	0.9
Client Avg	0.8808	0.8859	0.8895	**0.8914**	0.8882
Global	0.8964	0.8896	0.9019	**0.9028**	0.9001

Table 7. FedSM with different coefficient λ in retinal segmentation. Dice = average of disc and cup Dice coefficients.

In Table 5, we also validate that the threshold γ helps improve the performance of FedSM for the unseen data. For those unseen data with low confidence from the model selector, a larger γ increases the chance of the global model to be selected because maybe none of the personalized models is suitable. By choosing a proper γ, we can further improve the Dice of unseen clients 5 and 6 by 3%.

Validate SoftPull. Recall that our second motivation is to find a model generalizing well on the local data distribution even with insufficient local data. To achieve it we propose a new personalized FL optimization formulation with SoftPull to solve it. The Remark 1.1 of the theoretical analysis can be empirically validated by the fact that the best $\lambda = 0.7$ (closer to 1) for the retinal segmentation task with lower data similarity, and that the best $\lambda = 0.3$ (closer to $\frac{1}{K} = \frac{1}{6} = 0.17$) for the prostate segmentation task with higher data similarity.

Next, we will validate Remark 1.2 that a proper λ may lead to a convergence error, but in the meantime may improve the generalization by preventing overfitting the small local dataset with the help of other clients. We plot the 1D loss surface near the trained model by computing the loss along 10 randomly sampled unit vector directions (Figure 4), following existing works [18, 23]. It is interesting to see that local training overfits the training data and leads to a sharp local training optimum, which is known to generalize worse [18, 23, 51]. On the contrary, we observe an "over-regularization" effect for FedAvg as it has an even flatter training optimum than centralized training and a large convergence error (worse training loss), which also leads to a worse generalization performance. Indeed, averaging model in FedAvg can be regarded as a sort of implicit regularization. In comparison, SoftPull achieves a tunable flatness by choosing a proper λ. Even if it leads to a convergence error, it achieves generalization performance better than local training and comparable to centralized training.

4.3. Ablation Study

Personalization. We compare personalization methods in FedSM in Table 6, including (1) FT (local fine-tuning) [47], (2) APFL [10], (3) Per-FedAvg [11], and (4) Per-FedMe [45]. All methods' hyper-parameters are tuned for best results. SoftPull is the better interpolation method among them, outperforming APFL by 0.62% regarding the global Dice coefficient. It also outperforms the best counterpart by 0.44%.

Interpolation Coefficient λ. We explore different λ values of FedSM in Table 7 and $\lambda = 0.7$ performs the best.

5. Conclusion

In this work, we propose FedSM to close the generalization gap between FL and centralized training for medical image segmentation for the first time. The empirical study on real-world medical FL tasks validates our theoretical analysis and motivation to avoid the client drift issue.

References

[1] Nci_isbi dataset. https://www.cancerimagingarchive.net/. 5

[2] Promise12 dataset. https://promise12.grand-challenge.org/. 5

[3] Prostatex dataset. https://prostatex.grand-challenge.org/. 5

[4] Refuge dataset. https://refuge.grand-challenge.org/details/. 5

[5] 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*, 2020. 2

[6] Ahmed Almazroa, Sami Alodhayb, Essameldin Osman, Eslam Ramadan, Mohammed Hummadi, Mohammed Dlaim, Muhannad Alkatee, Kaamran Raahemifar, and Vasudevan Lakshminarayanan. Retinal fundus images for glaucoma analysis: the riga dataset. In *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*, volume 10579, page 105790B. International Society for Optics and Photonics, 2018. 5

[7] Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, Bram van Ginneken, et al. The medical segmentation decathlon. *arXiv preprint arXiv:2106.05735*, 2021. 5

[8] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27:1646–1654, 2014. 2

[9] Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*, 2018. 2

[10] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 4, 8

[11] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 8

[12] Francisco Fumero, Silvia Alayón, José L Sanchez, Jose Sigut, and M Gonzalez-Hernandez. Rim-one: An open retinal image database for optic nerve evaluation. In *2011 24th international symposium on computer-based medical systems (CBMS)*, pages 1–6. IEEE, 2011. 5

[13] Hongchang Gao, An Xu, and Heng Huang. On the convergence of communication-efficient local sgd for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence, Virtual*, pages 18–19, 2021. 1

[14] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020. 1

[15] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[16] Bin Gu, An Xu, Zhouyuan Huo, Cheng Deng, and Heng Huang. Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 1

[17] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2423–2432, 2021. 1

[18] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *arXiv preprint arXiv:1902.00744*, 2019. 8

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[20] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020. 1

[21] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 1

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 2

[23] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 8

[24] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019. 2

[25] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013. 2

[26] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 1

[27] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 2, 5, 6

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[29] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 1

[30] Guillaume Lemaître, Robert Martí, Jordi Freixenet, Joan C Vilanova, Paul M Walker, and Fabrice Meriaudeau.

Computer-aided detection and diagnosis for prostate cancer based on mono and multi-parametric mri: a review. *Computers in biology and medicine*, 60:8–31, 2015. 5

[31] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14)*, pages 583–598, 2014. 1

[32] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021. 2

[33] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 2, 6

[34] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020. 2

[35] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019. 2

[36] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021. 1, 2

[37] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. 3, 4

[38] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 6

[39] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625, 2019. 2

[40] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2020. 4

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 5

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 5

[43] J. Sivaswamy, S. R. Krishnadas, G. Datt Joshi, M. Jain, and A. U. Syed Tabish. Drishti-gs: Retinal image dataset for optic nerve head(onh) segmentation. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 53–56, April 2014. 5

[44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[45] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 8

[46] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 7611–7623, 2020. 2

[47] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019. 2, 8

[48] An Xu and Heng Huang. Double momentum sgd for federated learning. *arXiv preprint arXiv:2102.03970*, 2021. 1

[49] An Xu, Zhouyuan Huo, and Heng Huang. On the acceleration of deep learning model parallelism with staleness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2088–2097, 2020. 1

[50] An Xu, Zhouyuan Huo, and Heng Huang. Step-ahead error feedback for distributed training with compressed gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10478–10486, 2021. 1

[51] Guandao Yang, Tianyi Zhang, Polina Kirichenko, Junwen Bai, Andrew Gordon Wilson, and Chris De Sa. Swalp: Stochastic weight averaging in low precision training. In *International Conference on Machine Learning*, pages 7015–7024. PMLR, 2019. 8

[52] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021. 1

[53] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019. 2

[54] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020. 1

[55] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32:14774–14784, 2019. 1