Estimating before Debiasing: A Bayesian Approach to Detaching Prior Bias in Federated Semi-Supervised Learning

Guogang Zhu[1] , **Xuefeng Liu**[1,2] , **Xinghao Wu**[1] , **Shaojie Tang**[3] , **Chao Tang**[1] ,
Jianwei Niu[1,2*] and **Hao Su**[1]
[1]State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China
[2] Zhongguancun Laboratory, Beijing, China
[3]Jindal School of Management, The University of Texas at Dallas, Richardson, TX, USA
{buaa_zgg, liu_xuefeng, wuxinghao}@buaa.edu.cn, shaojie.tang@utdallas.edu, {sy2106322, niujianwei, bhsuhao}@buaa.edu.cn

Abstract

Federated Semi-Supervised Learning (FSSL) leverages both labeled and unlabeled data on clients to collaboratively train a model. In FSSL, the heterogeneous data can introduce prediction bias into the model, causing the model's prediction to skew towards some certain classes. Existing FSSL methods primarily tackle this issue by enhancing consistency in model parameters or outputs. However, as the models themselves are biased, merely constraining their consistency is not sufficient to alleviate prediction bias. In this paper, we explore this bias from a Bayesian perspective and demonstrate that it principally originates from label prior bias within the training data. Building upon this insight, we propose a debiasing method for FSSL named FedDB. FedDB utilizes the Average Prediction Probability of Unlabeled Data (APP-U) to approximate the biased prior. During local training, FedDB employs APP-U to refine pseudo-labeling through Bayes' theorem, thereby significantly reducing the label prior bias. Concurrently, during the model aggregation, FedDB uses APP-U from participating clients to formulate unbiased aggregate weights, thereby effectively diminishing bias in the global model. Experimental results show that FedDB can surpass existing FSSL methods. The code is available at https://github.com/GuogangZhu/FedDB.

1 Introduction

Federated Learning (FL) [McMahan *et al.*, 2017] is a distributed learning paradigm that can facilitate collaborative model training among multiple clients while preserving data privacy. Presently, most FL methods are confined to supervised learning (SL) settings, wherein it is presumed that each client maintains a fully labeled dataset. Nevertheless, in real-world applications, data labeling is notably laborious and time-consuming. Therefore, a more realistic case involves each client possessing a mix of unlabeled and labeled data. This specific scenario, known as Federated Semi-Supervised Learning (FSSL), has been explored in various studies [Jeong *et al.*, 2021; Lin *et al.*, 2021; Diao *et al.*, 2022] and is garnering increasing interest within the FL research community.

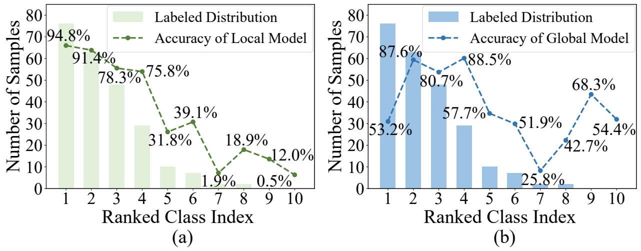

Figure 1: Class-wise test accuracy on a balanced test dataset, along with the labeled data distribution on an individual client. (a) Test accuracy of local model, (b) Test accuracy of global model. The class indexes are ranked based on the labeled data distribution.

In this study, we focus on an FSSL setting where the data on each client are class-imbalanced. Moreover, it is assumed that both intra-client and inter-client data heterogeneity exist. Specifically, intra-client data heterogeneity implies that both the labeled data and unlabeled data on an individual client originate from diverse distributions. Inter-client data heterogeneity means that the overall distributions across clients are non-independent and identically distributed (Non-IID).

In the described scenario, the model's prediction can skew to some certain classes during the training, i.e., prediction bias. Figure 1 presents the experimental results conducted in the above scenario, where the overall distributions of labeled and unlabeled data are balanced. It can be observed that due to class imbalance in the local client, the local model's predictions gradually skew towards the major classes in the local data. More importantly, this prediction bias cannot be alleviated after model aggregation, even if the overall distributions are balanced. Instead, it evolves into a different form of bias due to the influence from other clients. This bias can disrupt the pseudo-labeling process, further creating a 'vicious cycle' between pseudo-labeling and local model training.

Existing FSSL methods attribute the above issue to the di-

[*] Jianwei Niu is the corresponding author.

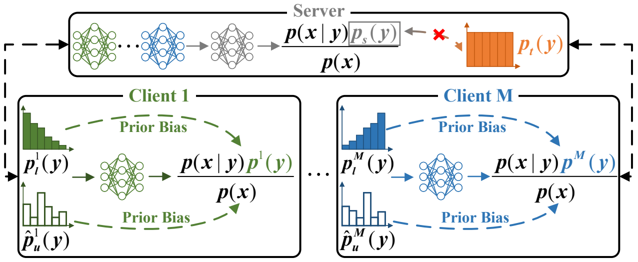

Figure 2: Prior bias in class-imbalanced FSSL.

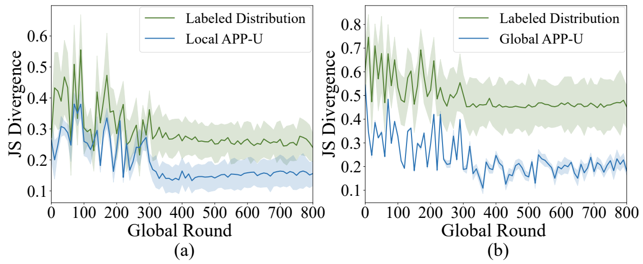

Figure 3: JS divergence between the ground truth bias and either the labeled data distribution or APP-U on clients. (a) Results on the local model, (b) Results on the global model.

vergence across clients caused by heterogeneous data and primarily address it by promoting consistency between model parameters or outputs [Zhang *et al.*, 2021; Jiang *et al.*, 2022; Liang *et al.*, 2022]. However, as both the local and global models are biased, merely constraining their consistency cannot fundamentally mitigate the model prediction bias.

In this paper, we delve into the essential reason for the prediction bias in FSSL from a Bayesian perspective. Based on Bayes' rule, the model prediction is as follows:

$$\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}) = \frac{\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})\boldsymbol{p}(\boldsymbol{y})}{\boldsymbol{p}(\boldsymbol{x})}, \tag{1}$$

where $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})$ is the model's prediction, $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y})$ is the class conditional likelihood, $\boldsymbol{p}(\boldsymbol{y})$ is the label prior. As shown in Figure 2, both the label prior of local labeled data (i.e., $\boldsymbol{p}_l(\boldsymbol{y})$) and unlabeled data (i.e., $\hat{\boldsymbol{p}}_u(\boldsymbol{y})$) are biased. Consequently, the model can gradually absorb these biases during training. These biases are eventually injected into the global model through model aggregation, causing its output priors $\boldsymbol{p}_s(\boldsymbol{y})$ to skew towards certain classes. When conducting inference on a balanced test dataset (i.e., $\boldsymbol{p}_t(\boldsymbol{y})$), the model may suffer severe performance degradation, as $\boldsymbol{p}_s(\boldsymbol{y}) \neq \boldsymbol{p}_t(\boldsymbol{y})$.

Nevertheless, $\boldsymbol{p}_s(\boldsymbol{y})$ is commonly challenging to estimate. On the one hand, in local clients, the ambiguity of pseudo-labels for unlabeled data makes the label prior bias during local training intractable. On the other hand, in the server, model aggregation combines influences from participating clients, further complicating the estimation of prior bias.

Taking the class-wise accuracy on a balanced test dataset as the ground truth for prior bias, we find that the Average Prediction Probability of Unlabeled Data (APP-U) serves as a robust metric to approximate this bias. Figure 3 illustrates the Jensen–Shannon (JS) divergence [Lin, 1991] between the ground truth bias and either the labeled data distribution or APP-U, where the solid line and shaded area represent the mean and range across clients, respectively. Interestingly, it reveals that for both the global and local models, prior bias does not consistently align with the labeled data distribution. Rather, it shows a stronger correlation with APP-U, indicating that APP-U can effectively quantify the prior bias.

Building upon the above insights, we introduce a hierarchical debiasing method for FSSL termed FedDB, to mitigate the prior bias at both the local training and global aggregation stages. During the local training, FedDB implements debiased pseudo-labeling (DPL) based on Bayes' theorem, with APP-U serving as the approximation of bias prior. This approach promotes a more balanced pseudo-labeling process for unlabeled data, substantially reducing the label prior bias during local training. At the global aggregation stage, FedDB utilizes APP-U from the participating clients to determine optimal aggregation weights. The above process, termed debiased model aggregation (DMA), effectively mitigates bias within the global model. It should be noted that DPL can be seamlessly integrated with FSSL methods that utilize pseudo-labeling with minimal cost. This demonstrates its substantial potential for practical application of FSSL.

The main contributions of this paper are as follows:

- We analyze the prediction bias in class-imbalanced FSSL from a Bayesian perspective.
- We propose FedDB, a Bayesian debiasing method for FSSL that uses APP-U as an approximation of prior bias.
- We conduct extensive experiments on multiple datasets to demonstrate the effectiveness of FedDB.

2 Related Work

2.1 Federated Learning

Data heterogeneity is a substantial challenge in FL, which can lead to considerable divergence across clients, thereby degrading the model performance [Zhao *et al.*, 2018; Li *et al.*, 2020a]. To address this issue, various strategies are explored, including reducing the divergence across local models [Li *et al.*, 2020b; Acar *et al.*, 2020; Karimireddy *et al.*, 2020], enhancing aggregation schemes [Wang *et al.*, 2020; Acar *et al.*, 2020; Reddi *et al.*, 2021], promoting representation consistency across clients [Tan *et al.*, 2022; Zhu *et al.*, 2023; Liao *et al.*, 2023], developing personalized models for individual clients [Collins *et al.*, 2021; Liu *et al.*, 2023]. However, these methods primarily focus on SL settings, which is impractical as data labeling is laborious and time-consuming.

2.2 Semi-Supervised Learning

SSL aims to mitigate the reliance on labeled data, which prompts various mechanisms to leverage the latent information within unlabeled data. Pseudo-labeling [Lee and others, 2013; Wang *et al.*, 2023], also known as self-training, involves assigning pseudo-labels to unlabeled samples with high confidence, enabling their incorporation into the training process. Consistency regularization [Miyato *et al.*, 2018] introduces arbitrary perturbations to unlabeled samples and promotes the consistent predictions between different views of unlabeled data. Additionally, hybrid methods that amalgamate these approaches are also developed, such as MixMatch [Berthelot *et al.*, 2019], FixMatch [Sohn *et al.*, 2020]. Recently, SSL has focused on class imbalance, leading to various studies such as class-rebalancing sampling [Wei *et al.*, 2021], and pseudo label sampling [Guo and Li, 2022]. However, simply combining these methods with FL is challenging, as they ignore the collaboration across clients.

2.3 Federated Semi-Supervised Learning

FSSL can be divided into three distinct scenarios [Bai *et al.*, 2023]: (1) **Labels-at-Partial-Clients**, where only a few clients have full labels, while the rest possess only unlabeled data [Liang *et al.*, 2022; Li *et al.*, 2023]; (2) **Labels-at-Server**, where labeled data are only available at the server, with local clients merely having unlabeled data [Zhang *et al.*, 2021; Jeong *et al.*, 2021; Diao *et al.*, 2022]; (3) **Labels-at-Clients**, where each client has mostly unlabeled data and a few labeled samples [Jeong *et al.*, 2021; Bai *et al.*, 2023].

This paper focuses on the **Labels-at-Clients** scenario. Currently, several works have been proposed for this scenario. For instance, SemiFed [Lin *et al.*, 2021] assigns pseudo-labels to unlabeled data only when multiple models provide consistent predictions. FedMatch [Jeong *et al.*, 2021] enforces prediction consistency across multiple models. However, these methods primarily concentrate on encouraging consistency across clients, overlooking the inherent prior biases within the model — a critical factor leading to performance degradation in FSSL with class imbalance.

3 Preliminary and Background

In this section, we present the notations used in this paper, followed by a detailed discussion of the framework of FSSL.

3.1 Problem Setting and Notation of FSSL

We focus on a FSSL setting for K-class classification task with totally M clients participating in the training. Each client m maintains a labeled dataset $\mathcal{D}_l^m = \{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^{N_l^m}$ and an unlabeled dataset $\mathcal{D}_u^m = \{(\boldsymbol{x}^n)\}_{n=1}^{N_u^m}$, where N_l^m and N_u^m are the counts of labeled and unlabeled samples, respectively (typically, $N_u^m \gg N_l^m$), $\boldsymbol{x}^n \in \mathcal{X} \subseteq \mathbb{R}^d$ is the input sampled from a d-dimensional space, $\boldsymbol{y}^n \in \mathcal{Y} \subseteq \{0,1\}^K$ is the one-hot label. For clarity, we sometimes omit the superscript denoting the client index in the following contents.

With a slight abuse of notation, we denote N_l^k and N_u^k as the numbers of samples in class k under $\mathcal{D}_l$ and $\mathcal{D}_u$ for an arbitrary client, i.e., $\sum_{k=1}^K N_l^k = N_l$ and $\sum_{k=1}^K N_u^k = N_u$. In this paper, we assume that both $\mathcal{D}_l$ and $\mathcal{D}_u$ exhibit class imbalance, that is, $\exists i, j \in \{1, 2, \dots, K\}$ for which the ratio $\frac{N_l^i}{N_l^j}$ is significantly greater than 1. In other words, the label prior distribution $\{p_l^1, \dots, p_l^K\}$ shifts from a uniform distribution $\{\frac{1}{K}\}^K$. This assumption is similarly applicable for $\mathcal{D}_u$.

Furthermore, we consider the setting that both intra-client and inter-client data heterogeneity exist in the FL system. Intra-client heterogeneity refers to the varied distributions of labeled and unlabeled data within a single client, that is, $\forall m \in \{1, 2, \dots, M\}, \mathcal{D}_u^m \neq \mathcal{D}_l^m$. Inter-client heterogeneity, on the other hand, pertains to the dissimilar mixture distributions of both labeled and unlabeled data across clients, i.e., $\forall i, j \in \{1, 2, \cdots M\}, i \neq j,$ it holds that $\mathcal{D}_l^i + \mathcal{D}_u^i \neq \mathcal{D}_l^j + \mathcal{D}_u^j$.

The final objective of FSSL is to learn a global model $f(\boldsymbol{x}; \boldsymbol{w}) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\boldsymbol{w}$ that can generalize well to a balanced test dataset whose label prior distribution is $\{\frac{1}{K}\}^K$. Given the input $\boldsymbol{x}^n$, we denote its corresponding output logits as $\boldsymbol{z}(\boldsymbol{x}^n) := f(\boldsymbol{x}^n; \boldsymbol{w})$, and the normalized prediction probability after softmax layer as $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x}^n) := \sigma(f(\boldsymbol{y}|\boldsymbol{x}^n; \boldsymbol{w}))$, where $\sigma(\cdot)$ is the softmax function. The detailed framework of FSSL is explained as follows.

3.2 Framework of FSSL

During each global round, the server first selects a random subset of clients $\mathcal{S}$ based on the activation rate C and broadcasts the global model $\boldsymbol{w}$ to these clients. Subsequently, these clients perform local training for E epochs using $\boldsymbol{w}$ as initial weights, resulting in the updated local model $\boldsymbol{w}_m$. Finally, the selected clients upload their local models $\boldsymbol{w}_m$ to the server for model aggregation. The training paradigms of labeled and unlabeled data in local clients are as follows.

For labeled data, the standard cross-entropy loss is applied to the weakly augmented version of samples to promote the discriminative objective, as shown below:

$$\mathcal{L}_s = \frac{1}{N_l} \sum_{n=1}^{N_l} \mathrm{H}(\boldsymbol{y}^n, \boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}^n))), \tag{2}$$

where $\alpha(\cdot)$ is the weak augmentation function, $\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}^n))$ is the prediction probability for $\alpha(\boldsymbol{x}^n)$, and $\mathrm{H}(\boldsymbol{p}_1, \boldsymbol{p}_2)$ is entropy between probability distributions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

For unlabeled data, the samples are pseudo-labeled using the trained model, after which they are incorporated into the training process. Specifically, for a given unlabeled sample $\boldsymbol{x}^n$, the model first generates the probability on its weakly augmented version. Then the pseudo-label is calculated by:

$$\hat{\boldsymbol{y}}^n = \arg\max(\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}^n))), \tag{3}$$

where $\arg\max(\cdot)$ is the function that converts a probability distribution into a one-hot label based on its maximum value.

To enhance the model generalization, the consistency loss is applied to unlabeled data by minimizing the entropy between the pseudo-label and the prediction of its strong augmented version.

During the training, only those unlabeled samples that exhibit high confidence are selected to participate in further training. Consequently, the overall optimization objective for the unlabeled data can be expressed as follows:

$$\mathcal{L}_u = \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbb{1}(\max(\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}^n))) \geq \tau) \cdot \tag{4}$$

$$\mathrm{H}(\hat{\boldsymbol{y}}^n, \boldsymbol{p}(\boldsymbol{y}|\mathcal{A}(\boldsymbol{x}^n))), \tag{5}$$

where τ is the threshold, $\mathbb{1}(\cdot)$ is the indicator function, $\mathcal{A}(\cdot)$ is the strong augmentation function.

The overall optimization objective of local training on clients is expressed as:

$$\mathcal{L} = \mathcal{L}_s + \lambda \mathcal{L}_u, \tag{6}$$

where λ is used to balance these two loss terms.

After local training, the selected clients send the latest local models to the server for model aggregation, as shown below:

$$\boldsymbol{w}^{t+1} = \frac{1}{|\mathcal{S}_t|}\sum_{m\in\mathcal{S}_t}\boldsymbol{\beta}_m \cdot \boldsymbol{w}_m^t, \tag{7}$$

where $|\mathcal{S}_t|$ is the number of selected clients in round t, $\boldsymbol{w}_m^t$ is the local model on client m in round t, $\boldsymbol{\beta}_m$ is the aggregate weight for $\boldsymbol{w}_m^t$, $\boldsymbol{w}^{t+1}$ is the global model in round $t+1$.

4 FedDB: Detaching Prior Bias in FSSL

This section details the framework of FedDB and its two key techniques: debiased pseudo-labeling (DPL) and debiased model aggregation (DMA).

4.1 Framework Overview of FedDB

Figure 4 illustrates the framework of FedDB. During the training, each global round consists of the following steps:

(1) The server selects a subset of clients for training and broadcasts the global model to these clients;

(2) The clients perform inference on unlabeled data and calculate APP-U to estimate the prior bias;

(3) The clients perform DPL on unlabeled data using APP-U;

(4) The clients train the model utilizing both labeled data and pseudo-labeled data;

(5) The clients upload local models and APP-U to the server. The server performs DMA using APP-U from clients;

(6) Repeating steps 1-5 until the global model converges.

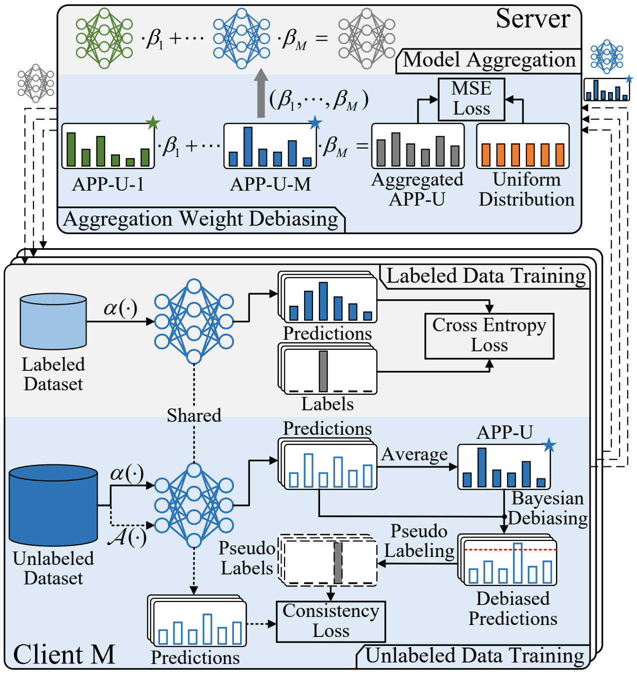

Figure 4: Framework overview of FedDB.

4.2 Prior Bias Estimation

In this paper, we consider an FSSL setting where both the labeled data and unlabeled data are class imbalanced. In such a case, the model's predictions can skew towards certain classes, owing to the biased label prior in the training data. This skew contradicts the training objective of FSSL, which is to achieve uniform performance across all classes.

To investigate the impact of class imbalance on model training in FSSL, we conduct preliminary experiments using the CIFAR10 dataset. We establish a scenario with 10 clients, each participating in model training in every round. The number of labeled and unlabeled samples is set to 4000 and 46000, respectively. The class imbalance is created by the Dirichlet distribution, as declared in Section 5.

As shown in Figure 1, both the local and global models exhibit a biased prediction towards certain classes. However, estimating the above bias in FSSL is challenging due to the data heterogeneity and imprecision in pseudo-labeling. By extensive experiments, we discover that the prior bias can be effectively approximated by the Average Prediction Probability on Unlabeled Data (APP-U). Specifically, for client m, the APP-U, denoted by $\overline{\boldsymbol{p}}_m$, can be calculated by:

$$\overline{\boldsymbol{p}}_m = \frac{\sum_{n=1}^{N_u^m}\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}_u^n))}{N_u^m}, \tag{8}$$

where N_u^m denotes the total number of unlabeled samples on client m, $\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}_u^n))$ is the prediction probability of the weak augmentation of sample $\boldsymbol{x}_u^n$.

We adopt JS divergence as a metric to quantify the disparity between two distributions. A larger JS divergence indicates a greater disparity between the distributions. Taking the class-wise accuracy on a balanced test dataset as the ground truth bias, we calculate the JS divergence between it and either APP-U or the labeled data distribution. As shown in Figure 3, for both local and global models, the JS divergence between APP-U and the ground truth is significantly smaller than that between the labeled data distribution and the ground truth. This demonstrates the effectiveness of APP-U as a metric for quantifying prior bias in FSSL.

4.3 Debiased Pseudo-Labeling

In this subsection, we detail the procedure of DPL. Given an FL model parameterized by $\boldsymbol{w}$, we first obtain the prediction probability $\boldsymbol{p}_s(\boldsymbol{y}|\boldsymbol{x})$ by applying a softmax function to unnormalized logits, as illustrated below:

$$\boldsymbol{p}_s(y|\boldsymbol{x}) = \frac{e^{\boldsymbol{z}(\boldsymbol{x})[y]}}{\sum_{k=1}^{K} e^{\boldsymbol{z}(\boldsymbol{x})[k]}}, \tag{9}$$

where $\boldsymbol{z}(\boldsymbol{x})[y]$ is the y-th unnormalized logit.

By applying the Bayes' theorem to $\boldsymbol{p}_s(y|\boldsymbol{x})$, we obtain:

$$\boldsymbol{p}_s(y|\boldsymbol{x}) = \frac{\boldsymbol{p}_s(y)\boldsymbol{p}_s(\boldsymbol{x}|y)}{\sum_{k=1}^{K}\boldsymbol{p}_s(k)\boldsymbol{p}_s(\boldsymbol{x}|k)}. \tag{10}$$

Due to the class imbalance in our FSSL settings, the prior distribution $\boldsymbol{p}_s(k)$, as outputted by the model, is biased towards certain majority classes. This leads to a biased prediction probability $\boldsymbol{p}_s(y|\boldsymbol{x})$, causing the model to be overconfident in these majority classes. The objective of DPL is to

Algorithm 1: DPL: Debiased Pseudo-labeling

```
Input: Confidence threshold $\tau$
Output: Debiased pseudo-labels $\hat{\boldsymbol{Y}}$, APP-U $\overline{\boldsymbol{p}}$
$\overline{\boldsymbol{p}} = \frac{\sum_{n=1}^{N_u} \boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}_u^n))}{N_u}$, $\hat{\boldsymbol{Y}} := \{\}$
for $n = 1, 2, ..., N_u$ do
    $\hat{\boldsymbol{p}}^n := \frac{\boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}_u^n))/\overline{\boldsymbol{p}}}{\sum_{k=1}^{K} \boldsymbol{p}(k|\alpha(\boldsymbol{x}_u^n))/\overline{\boldsymbol{p}}_k}$
    if $\max(\hat{\boldsymbol{p}}^n) \geq \tau$ then
        $\hat{\boldsymbol{Y}} := \hat{\boldsymbol{Y}} \oplus \arg\max(\hat{\boldsymbol{p}}^n)$
    else
        $\hat{\boldsymbol{Y}} := \hat{\boldsymbol{Y}} \oplus \{0\}^K$
Return $\hat{\boldsymbol{Y}}, \overline{\boldsymbol{p}}$
```

seek a conditional probability $\boldsymbol{p}_t(y|\boldsymbol{x})$ that is robust across all classes, given the estimation of the model's biased prior $\overline{\boldsymbol{p}}$, as defined in Eq. (8).

Following previous studies [Tian *et al.*, 2020; Kairouz *et al.*, 2021; Hong *et al.*, 2021], we assume that the class conditional likelihoods are the same in both the biased and debiased predictions, i.e., $\boldsymbol{p}_t(\boldsymbol{x}|y) = \boldsymbol{p}_s(\boldsymbol{x}|y)$. By rearranging Eq. (9) and Eq. (10), we have:

$$\begin{aligned}\ln(\boldsymbol{p}_t(y)\boldsymbol{p}_t(\boldsymbol{x}|y)) =& \boldsymbol{z}(\boldsymbol{x})[y] + \ln(\boldsymbol{p}_t(y)) - \ln(\boldsymbol{p}_s(y)) \\ &+ \ln(\textstyle\sum_{k=1}^{K} \boldsymbol{p}_s(k)\boldsymbol{p}_s(\boldsymbol{x}|k)) \\ &- \ln(\textstyle\sum_{k=1}^{K} e^{\boldsymbol{z}(\boldsymbol{x})[k]}).\end{aligned} \tag{11}$$

Recalling that:

$$\boldsymbol{z}(\boldsymbol{x})[y] = \ln \boldsymbol{p}_s(y|\boldsymbol{x}) + \ln \textstyle\sum_{k=1}^{K} \boldsymbol{p}_s(k)\boldsymbol{p}_s(\boldsymbol{x}|k). \tag{12}$$

We derive the following debiased posterior probability:

$$\begin{aligned}\boldsymbol{p}_t(y|\boldsymbol{x}) &= \frac{\boldsymbol{p}_t(y)\boldsymbol{p}_t(\boldsymbol{x}|y)}{\sum_{k=1}^{K} \boldsymbol{p}_t(k)\boldsymbol{p}_t(\boldsymbol{x}|k)} \\ &= \frac{\boldsymbol{p}_s(y|\boldsymbol{x})\boldsymbol{p}_t(y)/\boldsymbol{p}_s(y)}{\sum_{k=1}^{K} \boldsymbol{p}_s(k|\boldsymbol{x})\boldsymbol{p}_t(k)/\boldsymbol{p}_s(k)},\end{aligned} \tag{13}$$

where $\boldsymbol{p}_t(k)$ is a uniform distribution that is robust for all classes. By applying the estimated bias $\overline{\boldsymbol{p}}$ as the approximation of the prior bias $\boldsymbol{p}_s$, we can obtain the debiased prediction probability of unlabeled data as follows:

$$\hat{\boldsymbol{p}} = \frac{\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})/\overline{\boldsymbol{p}}}{\sum_{k=1}^{K} \boldsymbol{p}(k|\boldsymbol{x})/\overline{\boldsymbol{p}}_k}. \tag{14}$$

Intuitively, Eq. (14) serves as a regularization term that smooths the prediction probabilities of the majority classes and sharpens these of the minority classes, which can alleviate the prior bias introduced by the heterogeneous data. The detailed procedures of DPL are shown in Algorithm 1.

4.4 Debiased Model Aggregation

The objective of DMA is to computing aggregation weights that enable the model to perform uniformly across all classes. During each local updating round, the activated clients send their accumulated APP-U $\overline{\boldsymbol{p}}_m$ and their latest models $\boldsymbol{w}_m$ to the server. Then we can get the aggregated APP-U as follows:

Algorithm 2: DMA: Debiased Model Aggregation

```
Input: Local models$\{\boldsymbol{w}_m\}_{m=1}^{M}$, local APP-U $\{\overline{\boldsymbol{p}}_m\}_{m=1}^{M}$,
       updating epochs $E_{aggr}$, learning rate $\eta_{aggr}$
Output: Global weight $\boldsymbol{w}$
Initialize $\boldsymbol{\beta}$ as $\{\frac{1}{M}\}^M$
for $e = 1, 2, ..., E_{aggr}$ do
    $\overline{\boldsymbol{p}}_{aggr} \leftarrow \sum_{m=1}^{M} \boldsymbol{\beta}_m \overline{\boldsymbol{p}}_m$
    $\mathcal{L}_{aggr} = \sqrt{\sum_{m=1}^{M} (\overline{\boldsymbol{p}}_{aggr} - \boldsymbol{p}_t)^2}$
    $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \eta_{aggr} \nabla \mathcal{L}_{aggr}$
    $\boldsymbol{\beta} = \sigma(\boldsymbol{\beta})$
$\boldsymbol{w} \leftarrow \sum_{m=1}^{M} \boldsymbol{\beta}_m \boldsymbol{w}_m$
Return $\boldsymbol{w}$
```

$$\overline{\boldsymbol{p}}_{aggr} = \textstyle\sum_{m \in \mathcal{S}_t} \boldsymbol{\beta}_m \overline{\boldsymbol{p}}_m, \tag{15}$$

where $\boldsymbol{\beta}_m$ denotes the aggregation weight for client m. To achieve a more balanced model, we expect $\overline{\boldsymbol{p}}_{aggr}$ to be more uniform, leading to the following optimization objective:

$$\begin{aligned}&\min_{\boldsymbol{\beta}} \mathcal{L}_{aggr} = \sqrt{\textstyle\sum_{m=1}^{M} (\overline{\boldsymbol{p}}_{aggr} - \boldsymbol{p}_t)^2} \\ &\text{s.t.} \textstyle\sum_{m \in \mathcal{S}_t} \boldsymbol{\beta}_m = 1,\end{aligned} \tag{16}$$

where $\boldsymbol{p}_t = \{\frac{1}{K}\}^K$ is the uniform distribution over K classes, identical to the test dataset. In FedDB, we utilize the gradient descent algorithm to solve the above optimization problem.

After obtaining the aggregation weights $\boldsymbol{\beta}$, we aggregate client models and update the global model as follows:

$$\boldsymbol{w}^{t+1} = \textstyle\sum_{m \in \mathcal{S}_t} \boldsymbol{\beta}_m \cdot \boldsymbol{w}_m^t, \tag{17}$$

where $\boldsymbol{w}_m^t$ is the local model of client m at last round, $\boldsymbol{w}^{t+1}$ is the global model. $\boldsymbol{w}^{t+1}$ is then broadcast to the activated client for further updates. The processes of DMA and FedDB are presented in Algorithms 2 and 3, respectively.

5 Experiments

This section details the experimental results in various settings to demonstrate the effectiveness of FedDB.

5.1 Experimental Setup

Datasets. We evaluate FedDB on three benchmark datasets, including CIFAR10, SVHN, and CIFAR100. Initially, a balanced labeled dataset is separated from the original training dataset, with the residual data designated as the unlabeled dataset. When distributing these training data to clients, we sample data from a Dirichlet distribution $\boldsymbol{q} \sim \text{Dir}(\delta\boldsymbol{p})$, where $\boldsymbol{p}$ is the class-wise prior distribution and δ is a parameter that modulates the heterogeneity among clients. A higher value of δ correlates with reduced data heterogeneity. To enrich the unlabeled dataset, we add the samples from the labeled dataset to the unlabeled dataset after discarding their labels. We conduct experiments in IID setting and Non-IID settings with $\delta = \{0.1, 0.3\}$. In the IID setting, the total number of labeled samples is set to $4000, 1000, 10000$ for CIFAR10,

Algorithm 3: FedDB: Detaching Prior Bias in FSSL

```
Input: Client number $M$, client activate rate $C$, global
       rounds $T$, update epochs $E$ and $E_{aggr}$, learning rate
       $\eta$ and $\eta_{aggr}$, threshold $\tau$, unlabeled loss weight $\lambda$,
       momentum accumulation coefficient $\gamma$
Output: Global model $w^T$
Server executes:
Initialize $\boldsymbol{w}^0$
for $t = 1, 2, ..., T$ do
    $\mathcal{S}_t \leftarrow$ randomly select $M \cdot C$ clients
    for each client in $m \in S_t$ in parallel do
        $\boldsymbol{w}_m^t, \overline{\boldsymbol{p}}_m^t \leftarrow$ ClientUpdate($\boldsymbol{w}^{t-1}$)
    $\boldsymbol{w}^t \leftarrow$ DMA($\{\boldsymbol{w}_m^t\}_{m \in \mathcal{S}_t}, \{\overline{\boldsymbol{p}}_m^t\}_{m \in \mathcal{S}_t}, E_{aggr}, \eta_{aggr}$)
Return $\boldsymbol{w}^T$
ClientUpdate($\boldsymbol{w}^t$)
$\hat{\boldsymbol{Y}}, \overline{\boldsymbol{p}} \leftarrow$ DPL($\tau$)
for $e = 1, 2, ..., E$ do
    $\overline{\boldsymbol{p}}^e = \frac{\sum_{n=1}^{N_u} \boldsymbol{p}(\boldsymbol{y}|\alpha(\boldsymbol{x}_u^n))}{N_u}$
    $\mathcal{L}_s = \frac{1}{N_l} \sum_{n=1}^{N_l} \mathrm{H}(\boldsymbol{y}^n, p(\boldsymbol{y}|\alpha(\boldsymbol{x}_l^n)))$
    $\mathcal{L}_u = \frac{1}{N_u} \sum_{n=1}^{N_u} \mathbb{1}(\max(\hat{\boldsymbol{Y}}^n)) \geq \tau) \cdot$
                 $\mathrm{H}(\hat{\boldsymbol{Y}}^n, p(\boldsymbol{y}|\mathcal{A}(\boldsymbol{x}_u^n)))$
    $\mathcal{L} \leftarrow \mathcal{L}_s + \lambda \mathcal{L}_u$
    $\boldsymbol{w}^e \leftarrow \boldsymbol{w}^{e-1} - \eta \nabla \mathcal{L};\ \overline{\boldsymbol{p}} \leftarrow \gamma \overline{\boldsymbol{p}} + (1 - \gamma) \overline{\boldsymbol{p}}^e$
Return $\boldsymbol{w}^E, \overline{\boldsymbol{p}}$
```

SVHN and CIFAR100, respectively. For Non-IID setting, the total number of labeled data is set to 4000 for CIFAR10 and SVHN, and 10000 for CIFAR100. The test dataset from the original dataset is used for model evaluation.

Benchmark Methods. We compare FedDB against the following benchmark methods:

- **FedAvg** [McMahan *et al.*, 2017]: The FedAvg method is applied in a constrained scenario where each client utilizes only the small labeled dataset for training.
- **FixMatch** [Sohn *et al.*, 2020]: This method is a basic adaptation of FixMatch within FedAvg framework.
- **FedMatch** [Jeong *et al.*, 2021]: FedMatch introduces the inter-client consistency loss to maximize the agreement between local models.
- **FedRGD** [Zhang *et al.*, 2021]: It mitigates the model bias by reducing gradient divergence among clients.
- **SemiFL** [Diao *et al.*, 2022]: SemiFL adopts alternate training between server and clients. Here, we adopts its client-side training due to the lack of training samples on the server in our scenario.
- **Methods combining DPL.** We also conduct experiments that integrate DPL with benchmark methods. These hybrid methods are denoted as **Method-FedDPL**.

Implementation Details. We primarily follow the experimental settings adopted in prior works of FSSL [Jeong *et al.*, 2021]. There are a total of 100 clients participating in the training, with 10 active clients ($C = 0.1$) engaged in each global round. The local training epoch is set to $E = 5$ and the epoch for updating the model aggregation weights is set to $E_{aggr} = 100$. All experiments are executed for 800 global rounds. We employ Wide ResNet28x2 in our experiments. The SGD optimizer is adopted for model training, operating at learning rates $\eta = 0.03$ for local updating and $\eta_{aggr} = 1.0$ for aggregation, complemented by a momentum of 0.9. Due to the limited number of samples on clients, we feed all training data simultaneously to the model during local training. The confidence threshold for pseudo-labeling is set to $\tau = 0.95$. The data augmentation operation is consistent with those described in FixMatch [Sohn *et al.*, 2020]. All experiments are repeated for 4 times and we report the mean and standard deviation of the best accuracy during training.

5.2 Results on Benchmark Datasets

The experimental results are presented in Tables 1 - 3, where values inside the parentheses represent the mean, and values outside the parentheses represent the standard deviation of multiple experiments. It can be observed that with the same number of labeled samples, the accuracy of all methods decreases as δ decreases, demonstrating that data heterogeneity is a key factor harming model performance. FedAvg, despite its simplicity, serves as a reliable benchmark method, particularly as the dataset difficulty increases (e.g., CIFAR100). This issue is also noted by [Diao *et al.*, 2022]. This demonstrates that improperly incorporating unlabeled data into training can negatively impact the model's training. Compared with other FSSL methods, FedDB enhances test accuracy, demonstrating the effectiveness of FedDB in the FSSL scenario. The same conclusion can also be drawn from Figure 5.

Dataset	CIFAR10	SVHN	CIFAR100
FedAvg	58.42(0.61)	25.10(0.76)	32.00(0.80)
FixMatch	65.80(2.72)	87.44(1.35)	24.72(0.73)
FedMatch	39.63(1.66)	25.09(5.40)	9.44(0.66)
FedRGD	63.27(1.47)	81.04(2.43)	14.45(0.42)
SemiFL	57.24(7.96)	85.58(10.03)	22.61(3.07)
FixMatch-FedDPL	66.97(2.84)	88.00(0.67)	26.44(1.73)
FedMatch-FedDPL	43.06(3.16)	25.90(3.12)	9.47(0.79)
FedRGD-FedDPL	64.75(1.20)	81.24(5.36)	17.17(0.98)
SemiFL-FedDPL	68.46(3.61)	86.77(1.79)	27.67(0.89)
FedDB	67.32(2.31)	86.75(0.90)	26.71(0.87)

Table 1: Experimental results in the IID setting.

5.3 Effectiveness of DPL

As illustrated in Table 4, employing DPL results in substantial gains for FedDB. Furthermore, DPL can be regarded as a convenient plug-in that can be easily integrated into existing FSSL methods utilizing pseudo-labeling. As shown in Tables 1 - 3, introducing DPL to existing FSSL methods effectively enhances their performance. Figure 6 displays the accuracy of pseudo-labels during training. It indicates that DPL effectively enhances the accuracy of these pseudo-labels, which in turn benefits FSSL training. Figure 7 presents the ratio of pseudo-labeled samples in the unlabeled data. However,

Dataset	CIFAR10	SVHN	CIFAR100
FedAvg	47.72(1.95)	69.44(6.21)	31.34(0.36)
FixMatch	50.99(2.49)	86.61(0.19)	25.47(0.46)
FedMatch	38.64(2.49)	26.04(4.85)	8.77(0.57)
FedRGD	51.45(2.39)	86.89(3.21)	14.83(0.34)
SemiFL	50.07(1.05)	76.11(6.3)	26.40(0.81)
FixMatch-FedDPL	53.92(3.41)	85.87(0.51)	28.47(0.13)
FedMatch-FedDPL	39.17(2.10)	27.02(3.13)	8.87(0.11)
FedRGD-FedDPL	51.57(1.67)	87.00(1.31)	19.94(0.75)
SemiFL-FedDPL	55.42(2.57)	87.61(0.91)	28.29(0.73)
FedDB	55.00(1.17)	85.99(0.49)	29.28(0.51)

Table 2: Experimental results in the Non-IID setting with $\delta = 0.3$.

Dataset	CIFAR10	SVHN	CIFAR100
FedAvg	33.53(1.9)	32.21(1.52)	28.78(0.53)
FixMatch	35.14(1.53)	74.31(2.07)	25.90(1.06)
FedMatch	31.12(2.69)	12.66(3.34)	7.50(0.99)
FedRGD	35.33(3.73)	38.20(5.64)	18.04(1.59)
SemiFL	33.72(1.87)	72.76(6.19)	25.82(0.44)
FixMatch-FedDPL	37.13(3.22)	76.29(1.00)	27.76(0.85)
FedMatch-FedDPL	32.26(2.75)	16.94(1.28)	7.66(0.43)
FedRGD-FedDPL	35.59(3.49)	38.76(2.67)	18.98(0.58)
SemiFL-FedDPL	37.84(2.33)	74.54(7.51)	27.62(1.00)
FedDB	37.95(2.21)	76.20(1.31)	27.99(1.28)

Table 3: Experimental results in the Non-IID setting with $\delta = 0.1$.

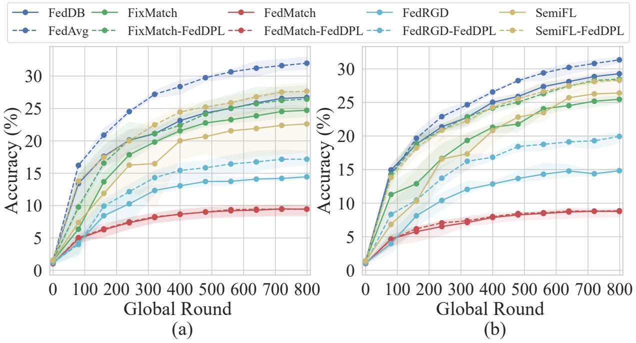

Figure 5: Convergence curve on CIFAR100. (a) IID, (b)Non-IID with $\delta = 0.3$.

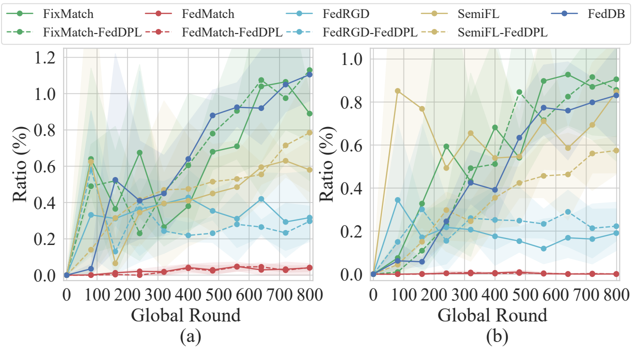

Figure 7: Ratio of unlabeled samples that are finally assigned with pseudo-labels on CIFAR100. (a) IID, (b)Non-IID with $\delta = 0.3$.

introducing DPL does not consistently improve the ratio of pseudo-labeled samples, as the model in FSSL is challenging to train, making it difficult for samples to be pseudo-labeled.

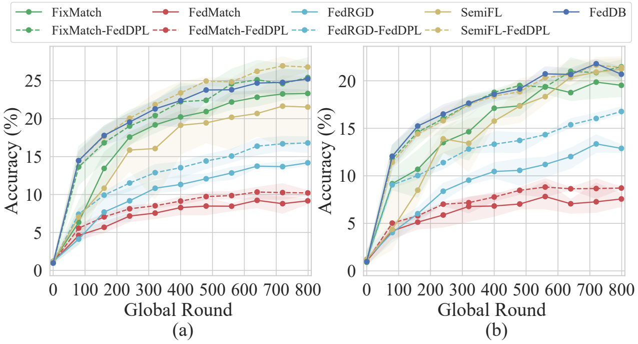

Figure 6: Accuracy of pseudo labels on CIFAR100. (a) IID, (b)Non-IID with $\delta = 0.3$.

5.4 Effectiveness of DMA

As shown in Table 4, DMA generally contributes positively to FedDB in most scenarios. However, its impact differs among various datasets. More specifically, DMA consistently results in improved outcomes on the CIFAR10 and CIFAR100 datasets. Conversely, on the SVHN dataset, DMA can lead to performance decline in certain scenarios. Upon detailed analysis, we ascribe this issue to the imbalanced distribution of the SVHN dataset, which contravenes the objective of FSSL that seeks for a balanced model.

	DPL	DMA	CIFAR10	SVHN	CIFAR100
IID	-	-	65.80(2.72)	87.44(1.35)	24.72(0.73)
	✓	-	66.97(2.84)	88.00(0.67)	26.44(1.73)
	✓	✓	67.32(2.31)	86.75(0.90)	26.71(0.87)
$\delta = 0.3$	DPL	DMA	CIFAR10	SVHN	CIFAR100
	-	-	50.99(2.49)	86.61(0.19)	25.47(0.46)
	✓	-	53.92(3.41)	85.87(0.51)	28.47(0.13)
	✓	✓	55.00(1.17)	85.99(0.49)	29.28(0.51)
$\delta = 0.1$	DPL	DMA	CIFAR10	SVHN	CIFAR100
	-	-	35.14(1.53)	74.31(2.07)	25.90(1.06)
	✓	-	37.13(3.22)	76.29(1.00)	27.76(0.85)
	✓	✓	37.95(2.21)	76.20(1.31)	27.99(1.28)

Table 4: Ablation studies on CIFAR10, SVHN, and CIFAR100.

6 Conclusion

In this paper, we propose FedDB to detach prior bias in FSSL with class imbalance. At the local training level, FedDB debiases the pseudo-labeling using APP-U based on Bayes' theorem, encouraging a more balanced training data during the training. At the global aggregation level, FedDB leverages APP-U across different clients to derive optimal aggregation weights, aiming to debias the global model. Extensive experiments have shown the effectiveness of FedDB.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 62372028 and 62372027.

References

[Acar *et al.*, 2020] 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.

[Bai *et al.*, 2023] Sikai Bai, Shuaicheng Li, Weiming Zhuang, Kunlin Yang, Jun Hou, Shuai Yi, Shuai Zhang, Junyu Gao, Jie Zhang, and Song Guo. Combating data imbalances in federated semi-supervised learning with dual regulators. *arXiv preprint arXiv:2307.05358*, 2023.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[Diao *et al.*, 2022] Enmao Diao, Jie Ding, and Vahid Tarokh. Semifl: Semi-supervised federated learning for unlabeled clients with alternate training. *Advances in Neural Information Processing Systems*, 35:17871–17884, 2022.

[Guo and Li, 2022] Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, pages 8082–8094. PMLR, 2022.

[Hong *et al.*, 2021] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6626–6636, 2021.

[Jeong *et al.*, 2021] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2021.

[Jiang *et al.*, 2022] Meirui Jiang, Hongzheng Yang, Xiaoxiao Li, Quande Liu, Pheng-Ann Heng, and Qi Dou. Dynamic bank learning for semi-supervised federated image diagnosis with class imbalance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 196–206. Springer, 2022.

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

[Karimireddy *et al.*, 2020] 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.

[Lee and others, 2013] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896, 2013.

[Li *et al.*, 2020a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

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

[Li *et al.*, 2023] Ming Li, Qingli Li, and Yan Wang. Class balanced adaptive pseudo labeling for federated semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16292–16301, 2023.

[Liang *et al.*, 2022] Xiaoxiao Liang, Yiqun Lin, Huazhu Fu, Lei Zhu, and Xiaomeng Li. Rscfed: random sampling consensus federated semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10154–10163, 2022.

[Liao *et al.*, 2023] Xinting Liao, Weiming Liu, Chaochao Chen, Pengyang Zhou, Huabin Zhu, Yanchao Tan, Jun Wang, and Yue Qi. Hyperfed: hyperbolic prototypes exploration with consistent aggregation for non-iid data in federated learning. *arXiv preprint arXiv:2307.14384*, 2023.

[Lin *et al.*, 2021] Haowen Lin, Jian Lou, Li Xiong, and Cyrus Shahabi. Semifed: Semi-supervised federated learning with consistency and pseudo-labeling. *arXiv preprint arXiv:2108.09412*, 2021.

[Lin, 1991] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[Liu *et al.*, 2023] Jiahao Liu, Jiang Wu, Jinyu Chen, Miao Hu, Yipeng Zhou, and Di Wu. Feddwa: personalized federated learning with dynamic weight adjustment. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3993–4001, 2023.

[McMahan *et al.*, 2017] 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.

[Miyato *et al.*, 2018] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[Reddi *et al.*, 2021] 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*, 2021.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

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

[Tian *et al.*, 2020] Junjiao Tian, Yen-Cheng Liu, Nathaniel Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior recalibration for imbalanced datasets. *Advances in Neural Information Processing Systems*, 33:8101–8113, 2020.

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

[Wang *et al.*, 2023] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, , Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. 2023.

[Wei *et al.*, 2021] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021.

[Zhang *et al.*, 2021] Zhengming Zhang, Yaoqing Yang, Zhewei Yao, Yujun Yan, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Improving semi-supervised federated learning by reducing the gradient diversity of models. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1214–1225. IEEE, 2021.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[Zhu *et al.*, 2023] Guogang Zhu, Xuefeng Liu, Shaojie Tang, and Jianwei Niu. Aligning before aggregating: Enabling communication efficient cross-domain federated learning via consistent feature extraction. *IEEE Transactions on Mobile Computing*, 2023.