SemiFL: Communication Efficient Semi-Supervised Federated Learning with Unlabeled Clients

Enmao Diao [1] **Jie Ding** [2] **Vahid Tarokh** [1]

Abstract

Federated Learning (FL) allows training machine learning models by using the computation and private data resources of many distributed clients such as smartphones and IoT devices. Most existing works on FL assume the clients have ground-truth labels. However, in many practical scenarios, clients do not have task-specific labels, e.g., due to a lack of expertise. This work considers a server that hosts a labeled dataset and wishes to leverage clients with unlabeled data for supervised learning. We propose a new FL framework referred to as SemiFL to address Semi-Supervised Federated Learning (SSFL). In SemiFL, clients have completely unlabeled data, while the server has a small amount of labeled data. SemiFL is communication efficient since it allows to train client-side unsupervised data for multiple local epochs. We demonstrate several strategies in SemiFL to enhance efficiency and prediction, and develop intuitions of why they work. In particular, we provide a theoretical analysis of the use of strong data augmentation for semi-supervised learning, which can be interesting in its own right.

Extensive empirical evaluations demonstrate that SemiFL can significantly improve the performance of a labeled server with unlabeled clients. Moreover, SemiFL can outperform many existing SSFL methods, and perform competitively with the state-of-the-art FL and centralized SSL results. For instance, in the standard communication efficient scenarios, SemiFL gives 93% accuracy on the CIFAR10 dataset with only 4000 labeled samples at the server. Such accuracy is 2% away from the result trained from 50000 fully labeled data, and improves 30% upon existing SSFL methods.

[1]Department of Electrical and Computer Engineering, Duke University, Durham, US [2]School of Statistics, University of Minnesota-Twin Cities, Minnesota, US. Correspondence to: Enmao Diao <enmao.diao@duke.edu>, Jie Ding <dingj@umn.edu>, Vahid Tarokh <vahid.tarokh@duke.edu>.

Preliminary work.

1. Introduction

For billions of users around the world, mobile devices and Internet of Things (IoT) devices are becoming common computing platforms (Lim et al., 2020). These devices produce a large amount of data that can be used to improve a variety of existing applications (Hard et al., 2018). Consequently, it has become increasingly appealing to process data and train models locally from privacy and economic standpoints. To address this, distributed machine learning framework of Federated Learning (FL) has been proposed (Konečnỳ et al., 2016; McMahan et al., 2017). This method aggregates locally trained model parameters in order to produce a global inference model without sharing private local data.

Most existing works of FL focus on supervised learning tasks assuming that clients have ground-truth labels. However, in many practical scenarios, most clients may not be experts in the task of interest to label their data. In particular, the private data of each client may be completely unlabeled. For instance, a healthcare system may involve a central hub ("server") with domain experts and a limited number of labeled data (such as medical records), together with many rural branches with non-experts and a massive number of unlabeled data. As another example, an autonomous driving startup ("server") may only afford beta-users assistance in labeling a road condition but desires to improve its modeling quality with the information provided by many decentralized vehicles that are not beta-users. The above scenarios naturally lead to the following important question. *How a server that hosts a labeled dataset can leverage clients with unlabeled data for a supervised learning task in the Federated Learning setting?*

In this work, we propose a new Federated Learning framework referred to as SemiFL to address the problem of Semi-Supervised Federated Learning (SSFL) as illustrated in Figure 1. Our solution is inspired by a series of recent Federated Learning and Semi-Supervised Learning (SSL) solutions. The key ingredient that enable SemiFL to utilize decentralized unsupervised data is that we alternate the training of labeled server and unlabeled clients to ensure that the quality of pseudo-labeling is highly maintained during the training. We perform extensive empirical experiments to evaluate SemiFL and compare it with various baselines and the state-of-the-art techniques. The results demonstrate that SemiFL can outperform existing SSFL methods and perform closely to the state-of-the-art of FL and centralized SSL results. In particular, we demonstrate the following.

- We identify the difficulty of combining the state-of-the-art Semi-Supervised Learning method FixMatch (Sohn et al., 2020a) with communication efficient Federated Learning method FedAvg (McMahan et al., 2017) methods. We also develop a theoretical analysis on strong data augmentation for SSL methods, the first in the literature to our best knowledge.
- To the best of our knowledge, we propose the first communication efficient SSFL method named SemiFL that can improve the performance of a labeled server using unlabeled clients, e.g., from 42% to 88% with 250 labeled data, and from 77% to 93% accuracy with 4000 labeled data on the CIFAR10 dataset.
- SemiFL achieves 30% improvement over the existing SSFL methods. Furthermore, SemiFL performs competitively with the state-of-the-art FL methods and centralized SSL methods. e.g., only 1% and 2% away from the state-of-the-art FL and centralized SSL results, respectively, for 4000 labeled data on the CIFAR10 dataset.

The outline of the rest of this paper is given below. In Section 2, we review the related work. In Section 3, we identify the problem of combining SSL with communication efficient FL methods, develop theoretical analysis on how the strong data augmentation can significantly improve the classification accuracy, and present the proposed SemiFL method with some intuitive explanations. In Section 4, we evaluate the empirical performance of the SemiFL. We make some concluding remarks in Section 5.

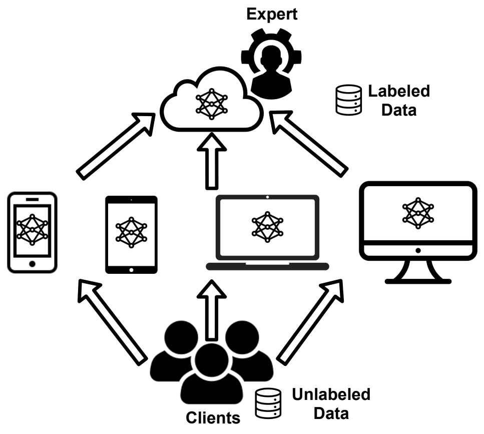

Figure 1. A resourceful server with labeled data can significantly improve its learning performance by working with distributed clients with unlabeled data without data sharing.

2. Related Work

Federated Learning The goal of Federated Learning is to scale and speed up the training of distributed models (Bonawitz et al., 2019; He et al., 2020). Communication efficiency, system heterogeneity, statistical heterogeneity, and privacy are all major issues in FL (Li et al., 2020b). To reduce the communication costs in FL, some studies propose using data compression techniques (Konečný et al., 2016; Alistarh et al., 2017; Ivkin et al., 2019), adding regularization terms for local optimization (Sahu et al., 2018; Acar et al., 2021), and developing FL counterparts of Batch Normalization (Hsieh et al., 2020; Li et al., 2021; Diao et al., 2021). Moreover, the use of local momentum and global momentum (Wang et al., 2019a) have been shown to facilitate faster convergence. In order to address potential system heterogeneity, asynchronous communication and active client sampling techniques have been developed (Bonawitz et al., 2019; Nishio & Yonetani, 2019). Statistical heterogeneity potentially poses yet another major challenge. Several methods have been proposed in order to address Non-IID data in FL, such as personalized local models (Liang et al., 2020), assisted learning (Xian et al., 2020), meta-learning (Jiang et al., 2019; Khodak et al., 2019), multi-task learning (Smith et al., 2017), transfer learning (Wang et al., 2019b; Mansour et al., 2020), knowledge distillation (Li & Wang, 2019), lottery ticket hypothesis (Li et al., 2020a), and masked cross-entropy (Diao et al., 2021) methods.

Semi-Supervised Learning Semi-Supervised Learning (SSL) refers to the general problem of learning with partially labeled data, especially when the amount of unlabeled data is much larger than that of the labeled data (Zhou & Li, 2005; Rasmus et al., 2015). The idea of self-training (namely to obtain artificial labels for unlabeled data from a pre-trained model) can be traced back to decades ago (Scudder, 1965; McLachlan, 1975), and has been applied to various domains such as language processing (McClosky et al., 2006), object detection (Rosenberg et al., 2005; Sohn et al., 2020b), image classification (Lee et al., 2013; Xie et al., 2020), and domain adaptation (Zou et al., 2018). Pseudo-labeling (Lee et al., 2013), a component of many recent SSL techniques (Miyato et al., 2018), is a form of entropy minimization (Grandvalet et al., 2005) by converting model predictions into hard labels. Consistency regularization (Bachman et al., 2014) refers to training models via minimizing the distance among stochastic outputs (Bachman et al., 2014; Rasmus et al., 2015). Various stochastic approaches have been proposed, such as the exponential moving average of model parameters (Tarvainen & Valpola, 2017), previous model checkpoints (Laine & Aila, 2016), stochastic regularization (Srivastava et al., 2014; Sajjadi et al., 2016; Laine & Aila, 2016), and adversarial perturbations (Miyato et al., 2018). A theoretical analysis of consistency regularization was recently developed in (Wei et al., 2021).

More recently, It has been demonstrated that the technique of strong data augmentation can lead to better outcomes (French et al., 2017; Xie et al., 2019; Berthelot et al., 2019b;a). Strongly augmented examples are frequently found outside of the training data distribution, which has been shown to benefit SSL (Dai et al., 2017). Noisy Student (Xie et al., 2020) has combined these strategies into a self-training framework, demonstrating outstanding performance on ImageNet with a large quantity of unlabeled data. Our work is based on the aforementioned SSL works, particularly the FixMatch (Sohn et al., 2020a) and ReMixMatch (Berthelot et al., 2019a).

Semi-Supervised Federated Learning (SSFL) Most existing FL works focus on supervised learning tasks, with clients having ground-truth labels. However, in many real-world scenarios, most clients are unlikely to be experts in the task of interest, an issue raised in a recent survey paper (Jin et al., 2020). In the research line of SSFL, a consistency loss based on the agreement among clients was developed in (Jeong et al., 2020). The paper (Albaseer et al., 2020) assumes that part of clients has unsupervised data and trains a convergent model at the server to label them. The paper (Itahara et al., 2020) considers using shared unlabeled data for Federated Distillation (Ahn et al., 2019; Sattler et al., 2020). Another related work (Zhang et al., 2020) trains and aggregates the model parameters of the labeled server and unlabeled clients in parallel. Applications of SSFL to specific applications can be found in, e.g., (Zhao et al., 2020; Yang et al., 2021).

In the standard communication efficient FL scenario (i.e.FedAvg (McMahan et al., 2017)), existing SSFL methods have difficulty in performing closely to the state-of-the-art centralized SSL methods (Jeong et al., 2020; Zhang et al., 2020; Long et al., 2020). This is somewhat surprising given that their underlying methods of training unlabeled data are similar. In fact, we will demonstrate that existing SSFL methods cannot outperform training with only labeled data.

The proposed method SemiFL not only outperforms existing SSFL methods but also performs competitively with the state-of-the-art FL methods and centralized SSL methods. SemiFL provides a large extension of FL to many practical applications where clients cannot access annotated data while the server can label data with experts.

3. Method

3.1. Problem

In a supervised learning classification task, we are given a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, where x_i is a feature vector, y_i is an one-hot vector representing the class label in a K-class classification problem, and N is the number of training examples. In a Semi-Supervised Learning classification task, we have two datasets, namely a supervised dataset $\mathcal{S}$ and an unsupervised dataset $\mathcal{U}$. Let $\mathcal{S} = \{x_s^i, y_s^i\}_{i=1}^{N_\mathcal{S}}$ be a set of $N_\mathcal{S}$ labeled data observations, and $\mathcal{U} = \{x_u^i\}_{i=1}^{N_\mathcal{U}}$ be a set of $N_\mathcal{U}$ unlabeled observations (without the corresponding true label y_u^i). It is often interesting to study the case where $N_\mathcal{S} \ll N_\mathcal{U}$.

In this work, we focus on Semi-Supervised Federated Learning (SSFL) with unlabeled clients as illustrated in Figure 1. Assume that there are M clients and let $x_{u,m}$ denote the set of unsupervised data available at client $m = 1, 2, \cdots, M$. Similarly, let (x_s, y_s) denote the set of labeled data available at the server. The server model is parameterized by model parameters W_s. The client models are parameterized respectively by model parameters $\{W_{u,1}, \ldots, W_{u,M}\}$. We assume that all models share the same model architecture, denoted by $f : (x, w) \mapsto f(x, w)$, which maps an input x and parameters W to a vector on the K-dimensional simplex, e.g., using softmax function applied to model outputs.

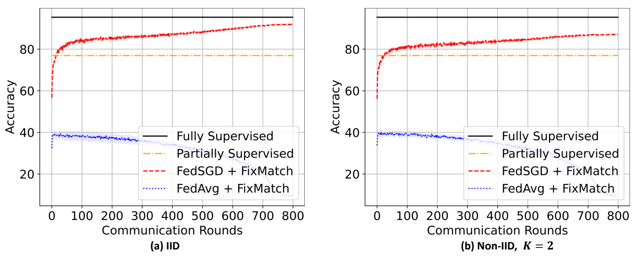

Figure 2. Results of CIFAR10 dataset with (a) IID and (b) Non-IID, $K = 2$ data partition and $N_\mathcal{S} = 4000$ with a vanilla combination of SSL and FL methods. The "Fully Supervised" and "Partially Supervised" refer to training a centralized model with full and 4000 labeled data respectively.

For communication efficient FL, local clients can train multiple epochs before model aggregation as suggested in FedAvg (McMahan et al., 2017). As shown in Figure 2, SSL methods such as FixMatch can only work with FedSGD, which requires batch-wise gradient aggregation and is not communication efficient. This is because SSL methods, such as FixMatch and MixMatch, synchronize the training of supervised and unsupervised data for every batch (Sohn et al., 2020a; Berthelot et al., 2019b). Thus, it is not straightforward how we can combine the SSL method and communication efficient FL method. To understand the bottleneck of this vanilla combination, we need to understand better why the state-of-the-art centralized SSL methods work.

3.2. Theoretical Analysis of Strong Data augmentation for SSL

To provide further insights into SSL, we develop a theoretical analysis of the strong data augmentation, which is a critical component of the state-of-the art SSL method

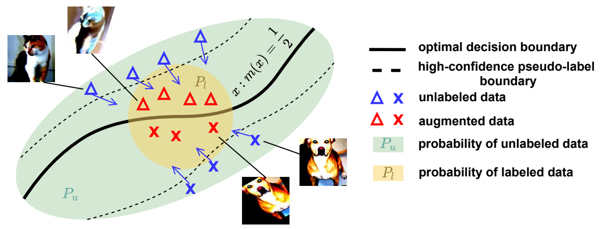
$x : m(x) = \frac{1}{2}$
P_l
P_u
optimal decision boundary
high-confidence pseudo-label boundary
unlabeled data
augmented data
P_u probability of unlabeled data
P_l probability of labeled data

Figure 3. Illustration of the strong data augmentation-based SSL. We pick up an unlabeled point ($X \sim \mathbb{P}_{\mathrm{u}}$) with a high-confidence pseudo-label, obtain its hard-thresholded label ($\hat{Y}$, which is believed to be close to the ground-truth), maneuver X into $\tilde{X}$ (which is believed to represent the test distribution $\mathbb{P}_{\mathrm{l}}$ to some extent), and then treat $(\hat{Y}, \tilde{X})$ as labeled data for training. Consequently, reliable task-specific information exhibited from unlabeled data can be transmitted to data regimes that may have been insufficiently trained with labeled data. Note that $\mathbb{P}_{\mathrm{l}}$ denotes the labeled data distribution as well as the out-sample test data distribution (used to evaluate the learning performance). The above ideas are theoretically formalized in Subsection 3.2 and Appendix D.

FixMatch (Sohn et al., 2020a) and SemiFL, and can be interesting in its own right. Intuitively, strong augmentation is a process that maps a data point (e.g., an image) from high quality to relatively low grade unilaterally. The low-quality data and their high-confidence pseudo-labels are then used for training so that there are sufficient "observations" in the data regime insufficiently covered by labeled data.

Our theory is based on an intuitive "adequate transmission" assumption, which basically means that the distribution of augmented data from high-confidence unlabeled data can adequately cover the data regime of interest during prediction. Consequently, reliable information exhibited from unlabeled data can be "transmitted" to data regimes that may have been insufficiently trained with labeled data, as illustrated in Figure 3. Instead of studying SSL in full generality, we restrict our attention to a class of nonparametric kernel-based classification learning (Audibert & Tsybakov, 2005; Kohler & Krzyzak, 2007; Devroye et al., 2013) and derive analytically tractable statistical risk-rate analysis. More detailed background and technical details are included in the Appendix. We provide a simplified statement as follows.

Theorem 1 *Under suitable assumptions, an SSL classifier $\hat{C}^{ssl}$ trained from n_u unlabeled data and the strong data augmentation technique has a statistical risk bound at the order of $\mathcal{R}(\hat{C}^{ssl}) \sim n_u^{-q(\alpha+1)/\{q(\alpha+3+\rho)+d\}}$ where d, q, α, ρ are constants that describe the data dimension, smoothness of the conditional distribution function $(Y \mid X)$, class separability (or task difficulty), and inadequacy of transmission, respectively. The smaller ρ, the better risk bound. Moreover, suppose that $\hat{C}^l$ is the classifier trained from n_l labeled data, where $n_l \sim n_u^{\zeta}$, $\zeta \in (0, 1)$. It can be verified that the bound of $\mathcal{R}(\hat{C}^u)$ is much smaller than that of $\mathcal{R}(\hat{C}^l)$ when $\zeta < \frac{q(\alpha+3)+d}{q(\alpha+3+\rho)+d}$. This provides an insight into the critical region of n_u where significant improvement can be made from unlabeled data.*

3.3. Alternate Training

As depicted in Figure 4(a), existing SSFL works follows the state-of-the-art SSL methods to synchronize the training of supervised and unsupervised data (Sohn et al., 2020a; Berthelot et al., 2019b), i.e. FedMatch (Jeong et al., 2020) and FedRGD (Zhang et al., 2020) adopt a vanilla combination of FixMatch and FedAvg. In particular, the vanilla method trains and aggregates the server's model parameters trained from labeled data and clients' model parameters trained from unlabeled data at each communication round in parallel. Moreover, it generates pseudo-labels for each batch of unlabeled data with the local training model. However, existing results (Jeong et al., 2020; Zhang et al., 2020) indicate that this vanilla combination has difficulty in performing close to the state-of-the-art centralized SSL methods, even if the unlabeled clients are trained with the aforementioned SSL methods. To understand the bottleneck of this vanilla extension, we need to intuitively clarify the reason that the centralized SSL methods work.

Pseudo-labeling is widely used for labeling unlabeled data in SSL methods (Lee et al., 2013; Xie et al., 2019; Sohn et al., 2020a). However, the quality (Accuracy) of those pseudo-labels can be low, especially at the beginning of the training. In this light, several papers (Xie et al., 2019; Sohn et al., 2020a) propose to hard-threshold or sharpen the pseudo-labels to improve the quantity of accurately labeled pseudo-labels. The problem with hard thresholding is that the data samples satisfying the confidence threshold have a small training loss. Therefore, the model cannot be significantly improved as it already performs well on the data above the threshold. To address this issue, we can use strong data augmentation (Dai et al., 2017; Sohn et al., 2020a) to generate data samples that have larger training loss. In summary, a successful SSL method must be able to generate more and more high-quality pseudo-labels during training, while the corresponding data used for training the model must have a larger loss gap than that of the original data.

However, in the communication efficient FL setting, we cannot guarantee an increase in the quality of pseudo-labels during training because we allow local clients to train multiple epochs, potentially deteriorating the performance. Furthermore, the aggregation of a server model trained with ground-truth labels and a subset of client models trained with pseudo-labels does not constantly improve the performance of the global model over the previous communication round. A poorly aggregated model of the last round results in worse quality pseudo-labels. Subsequently, the performance of the aggregated model degrades at the next round.

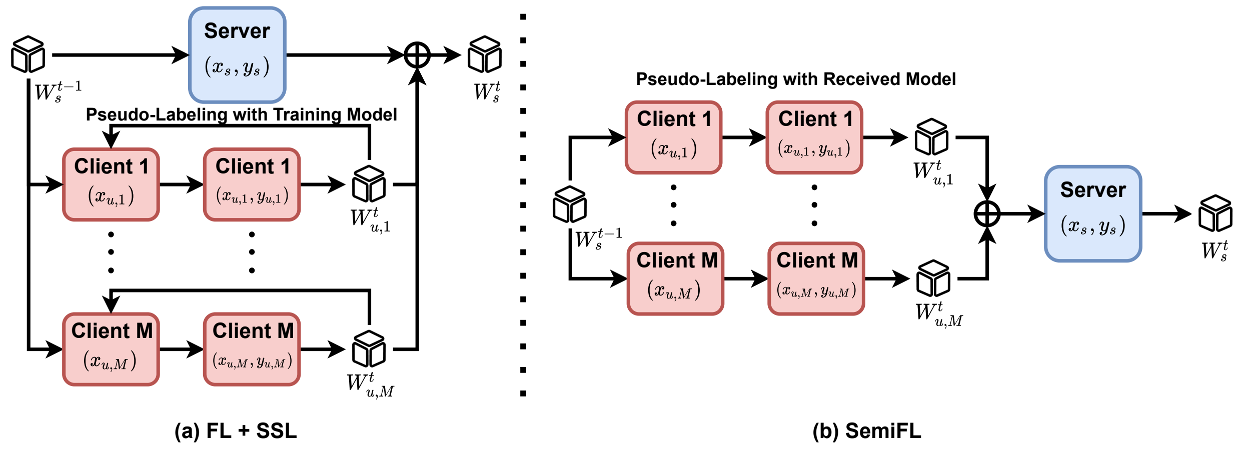

Figure 4. An illustration of (a) vanilla combination of FL and SSL, and (b) SemiFL. (a) The vanilla combination trains and aggregates server and clients models in parallel and generates pseudo-labels with the training models for each batch of unlabeled data. (b)SemiFL updates the aggregated client model with labeled data (also named "Fine-tuning") and generates pseudo-labels only once upon receiving the model from the server.

In order to maintain and improve the quality of our generated pseudo-labels during training, we propose to train the labeled server and unlabeled clients alternatively rather than in parallel, as illustrated Figure 4(b). In particular, our approach consists of two important components:

Fine-Tuning with Labeled Data At each round, the server will retrain the aggregated model with the labeled data. In this way, the server can provide a comparable or better model than the previous round for the active clients at the next round to generate pseudo-labels. Then, the quality of generated pseudo-labels can stably increase.

Pseudo-Labeling with Received Model We can label the unlabeled data once the active clients at the next round immediately receive the model from the server. On the contrary, the vanilla extension following centralized SSL methods labels every batch of data during training of unlabeled clients. The quality of generated pseudo-labels will gradually degrade during the training of local clients. Our proposed approach ensures that the clients can continually generate better quality pseudo-labels during training. Our experimental studies show that the proposed method can significantly improve the performance of the labeled server and performs competitively even with the state-of-the-art FL and centralized SSL methods. The limitation of our approach is that we need to update the aggregated client model with labeled data from the server, which will delay the computation time. We will conduct ablation studies on each component of alternative training.

3.4. The SemiFL Algorithm

We summarize the pseudo-code of the proposed solution in Algorithm 1. At each iteration t, the server will first update the model with the standard supervised loss L_s for local epochs E with data batch (x_b, y_b) of size B_s randomly split from the supervised dataset $\mathcal{D}_s$, using

$$L_s = \ell(f(\alpha(x_b), W_s), y_b), \quad W_s = W_s - \eta \nabla_W L_s, \quad (1)$$

where $\alpha(\cdot)$ represents a weak data augmentation, such as random horizontal flipping and random cropping, that maps one image to another one. Subsequently, the server updates the static Batch Normalization (sBN) statistics (Diao et al., 2021) (which is discussed in Appendix B). Next, the server distributes server model parameters W_s to a subset of clients. We denote the proportion of active clients at each communication round t as activity rate $C_t \in (0, 1]$. Without loss of generality, we assume that $C_t = C$ is a constant over time. After each active local client, say client m, receives the transmitted W_s, it generates pseudo-labels $y_{u,m}$ as follows:

$$W_{u,m} \leftarrow W_s, \quad y_{u,m} = f(\alpha(x_{u,m}), W_{u,m}). \quad (2)$$

Each local client will construct a high-confidence dataset $\mathcal{D}_{u,m}^{\text{fix}}$ inspired by FixMatch (Sohn et al., 2020a) at each iteration t, defined as:

$$\mathcal{D}_{u,m}^{\text{fix}} = \{(x_{u,m}, y_{u,m}) \ \texttt{with} \ \max(y_{u,m}) \geq \tau\}. \quad (3)$$

for a global confidence threshold $0 < \tau < 1$ pre-selected by all clients. If for some client m, we have $\mathcal{D}_{u,m}^{\text{fix}} = \emptyset$ then it will stop and refrain from transmission to the server. Otherwise, we will sample with replacement to construct a dataset inspired by MixMatch (Berthelot et al., 2019b). In other words,

$$\mathcal{D}_{u,m}^{\text{mix}} = \texttt{Sample} \ |\mathcal{D}_{u,m}^{\text{fix}}| \quad (4)$$
$$\texttt{with replacement}\{(x_{u,m}, y_{u,m})\},$$

where $|\mathcal{D}_{u,m}^{\text{fix}}|$ denotes the number of elements of $\mathcal{D}_{u,m}^{\text{fix}}$. Thus $|\mathcal{D}_{u,m}^{\text{mix}}| = |\mathcal{D}_{u,m}^{\text{fix}}|$. Subsequently, client m trains its

Algorithm 1 SemiFL: Semi-Supervised Federated Learning with Unlabeled Clients

```
Input: Unlabeled data $x_{u,1:M}$ distributed on $M$ local clients, activity rate $C$, the number of communication rounds $T$,
       the number of local training epochs $E$, server and client respective batch sizes $B_s$ and $B_m$, local learning rate $\eta$,
       server model parameterized by $W_s$ client models parameterized by $\{W_{u,1},\dots,W_{u,M}\}$, weak data augmentation
       function $\alpha(\cdot)$, strong data augmentation function $\mathcal{A}(\cdot)$, confidence threshold $\tau$, Mixup hyper-parameter $a$, loss
       hyperparameter $\lambda$, common model architecture function $f(\cdot)$
System executes:
    for each communication round $t = 1, 2, \dots T$ do
        $W_s^t \leftarrow$ ServerUpdate$(x_s, y_s, W_s^t)$
        Update the sBN statistics
        $S_t \leftarrow \max(\lfloor C \cdot M \rfloor, 1)$ active clients uniformly sampled without replacement
        for each client $m \in S_t$ in parallel do
            Distribute server model parameters to local client $m$, namely $W_{u,m}^t \leftarrow W_s^t$
            $W_{u,m}^t \leftarrow$ ClientUpdate$(x_{u,m}, W_{u,m}^t)$
        end
        Receive model parameters from $M_t$ clients, and calculate $W_s^t = M_t^{-1} \sum_{m=1}^{M_t} W_{u,m}^t$
    end
    $W_s^T \leftarrow$ ServerUpdate$(x_s, y_s, W_s^T)$
    Update the sBN statistics
ServerUpdate $(x_s, y_s, W_s)$:
    Construct supervised dataset $\mathcal{D}_s = (x_s, y_s)$
    for each local epoch $e$ from 1 to $E$ do
        $\mathcal{B}_s \leftarrow$ Randomly split local data $\mathcal{D}_s$ into batches of size $B_s$
        for batch $(x_b, y_b) \in \mathcal{B}_s$ do
            $L_s \leftarrow \ell(f(\alpha(x_b), W_s), y_b)$
            $W_s \leftarrow W_s - \eta \nabla_W L_s$
        end
    end
    Return $W_s$
ClientUpdate $(x_{u,m}, W_{u,m})$:
    Generate pseudo-label with weakly augmented data $\alpha(x_{u,m})$, namely $y_{u,m} = f(\alpha(x_{u,m}), W_{u,m})$
    Construct FixMatch dataset, namely $\mathcal{D}_{u,m}^{\text{fix}} = \{(x_{u,m}, y_{u,m}) \ \texttt{with} \ \max(y_{u,m}) \geq \tau\}$
    If $\mathcal{D}_{u,m}^{\text{fix}} = \emptyset$ then Stop. Return.
    Construct an equal-size MixMatch dataset, namely
    $\mathcal{D}_{u,m}^{\text{mix}} = \texttt{Sample} \ |\mathcal{D}_{u,m}^{\text{fix}}| \ \texttt{with replacement}\{(x_{u,m}, y_{u,m})\}$
    for each local epoch $e$ from 1 to $E$ do
        $\mathcal{B}_{u,m}^{\text{fix}}, \mathcal{B}_{u,m}^{\text{mix}} \leftarrow$ Randomly split local data $\mathcal{D}_{u,m}^{\text{fix}}, \mathcal{D}_{u,m}^{\text{mix}}$ into batches of size $B_m^{\text{fix}}, B_m^{\text{mix}}$
        for batch $(x_b^{\text{fix}}, y_b^{\text{fix}}), (x_b^{\text{mix}}, y_b^{\text{mix}}) \in \mathcal{B}_{u,m}^{\text{fix}}, \mathcal{B}_{u,m}^{\text{mix}}$ do
            $\lambda_{\text{mix}} \sim \text{Beta}(a, a)$
            $x_{\text{mix}} \leftarrow \lambda_{\text{mix}} x_b^{\text{fix}} + (1 - \lambda_{\text{mix}}) x_b^{\text{mix}}$
            $L_{\text{fix}} \leftarrow \ell(f(\mathcal{A}(x_b^{\text{fix}}), W_{u,m}), y_b^{\text{fix}})$
            $L_{\text{mix}} \leftarrow \lambda_{\text{mix}} \cdot \ell(f(\alpha(x_{\text{mix}}), W_{u,m}), y_b^{\text{fix}}) + (1 - \lambda_{\text{mix}}) \cdot \ell(f(\alpha(x_{\text{mix}}), W_{u,m}), y_b^{\text{mix}}))$
            $W_{u,m} \leftarrow W_{u,m} - \eta \nabla_W (L_{\text{fix}} + \lambda \cdot L_{\text{mix}})$
        end
    end
    Return $W_{u,m}$ and send it to the server
```

local model for E epoch to speed up convergence (McMahan et al., 2017). For each local training epoch of the client m, it randomly splits local data $\mathcal{D}_{u,m}^{\text{fix}}, \mathcal{D}_{u,m}^{\text{mix}}$ into batches $\mathcal{B}_{u,m}^{\text{fix}}, \mathcal{B}_{u,m}^{\text{mix}}$ of size B_m. For each batch iteration, as in (Zhang et al., 2017), client m constructs Mixup data from one particular data batch $(x_b^{\text{fix}}, y_b^{\text{fix}}), (x_b^{\text{mix}}, y_b^{\text{mix}})$ by

$$\lambda_{\text{mix}} \sim \text{Beta}(a, a), \quad x_{\text{mix}} \leftarrow \lambda_{\text{mix}} x_b^{\text{fix}} + (1 - \lambda_{\text{mix}}) x_b^{\text{mix}},$$

where a is the Mixup hyperparameter. Next, client m defines the "fix" loss L_{fix} (Sohn et al., 2020a) and "mix" loss

L_{mix} (Berthelot et al., 2019a) by

$$\begin{aligned} L_{\text{fix}} &= \ell(f(\mathcal{A}(x_b^{\text{fix}}), W_{u,m}), y_b^{\text{fix}}), \\ L_{\text{mix}} &= \lambda_{\text{mix}} \cdot \ell(f(\alpha(x_{\text{mix}}), W_{u,m}), y_b^{\text{fix}}) \\ &\quad + (1-\lambda_{\text{mix}}) \cdot \ell(f(\alpha(x_{\text{mix}}), W_{u,m}), y_b^{\text{mix}})). \end{aligned} \tag{5}$$

Here, $\mathcal{A}$ represents a strong data augmentation mapping, e.g., the RandAugment (Cubuk et al., 2020) used in our experiments, and ℓ is often the cross entropy loss for classification tasks. Finally, client m performs a gradient descent step with

$$W_{u,m} = W_{u,m} - \eta \nabla_W (L_{\text{fix}} + \lambda \cdot L_{\text{mix}}), \tag{6}$$

where $\lambda > 0$ is a hyperparameter set to be one in our experiments. After training for E local epochs, client m transmits $W_{u,m}$ to the server.

Without loss of generality assume that clients $1, 2, \cdots, M_t$ have sent their models to the server at time t. The server then aggregates client model parameters $\{W_{u,1}, \ldots, W_{u,M_t}\}$ by $W_s = M_t^{-1} \sum_{m=1}^{M_t} W_{u,m}$ (McMahan et al., 2017). This process is then repeated for multiple communication rounds T. After the training is finished, the server will further fine-tune the aggregated model by additional training with the server's supervised data using its supervised loss L_s. Finally, it will update the sBN statistics one final time.

4. Experiments

Experimental Setup To evaluate our proposed method, we conduct experiments with CIFAR10, SVHN, and CIFAR100 datasets (Netzer et al., 2011; Krizhevsky et al., 2009). To compare our method with existing FL and SSFL methods, we follow the standard communication efficient FL setting, which was originally used in FedAvg (McMahan et al., 2017) and widely adopted by following works, such as (Liang et al., 2020; Acar et al., 2021; Diao et al., 2021). We have 100 clients throughout our experiments, and the activity rate per communication round is $C = 0.1$. For IID data partition, we uniformly assign the same number of data examples to each client. For a balanced Non-IID data partition, we make sure each client has data at most from K classes, and the sample size of each class is the same. We set $K = 2$ because it is the most label-skewed case for classification, and it has been evaluated in (Liang et al., 2020; Acar et al., 2021; Diao et al., 2021). For unbalanced Non-IID data partition, we sample data for each client from a Dirichlet distribution $\text{Dir}(\alpha)$ (Hsu et al., 2019; Acar et al., 2021). As $\alpha \to \infty$, this reduces to IID data partition. We perform experiments with $\alpha = \{0.1, 0.3\}$.

To compare our method with the state-of-the-art SSL methods, we follow the experimental setup in (Sohn et al., 2020a). We use Wide ResNet28x2 (Zagoruyko & Komodakis, 2016) as our backbone model for CIFAR10 and SVHN datasets and Wide ResNet28x2 for CIFAR100 datasets throughout our experiments. The number of labeled data at the server for CIFAR10, SVHN, and CIFAR100 datasets $N_{\mathcal{S}}$ are $\{250, 4000\}$, $\{100, 2500\}$, and $\{2500, 10000\}$ respectively. We conduct four random experiments for all the datasets with different seeds, and the standard errors are shown inside the parentheses for tables and by error bars in figures. We demonstrate our experimental results in Table 1 and the learning curves of CIFAR10, SVNH, and CIFAR100 datasets in Figure 5, 6, and 7. Further details are included in the Appendix.

Comparison with SSL Methods We demonstrate the results of Fully Supervised and Partially Supervised cases and existing SSL methods for comparison in Table 1. The Fully Supervised case refers to all data being labeled, while in the Partially Supervised case, we only train the model with the partially labeled $N_{\mathcal{S}}$ data. Our results significantly outperform the Partially Supervised case. In other words, SemiFL can substantially improve the performance of a labeled server with unlabeled clients in a communication efficient scenario. For IID data partition, our method performs competitively with the state-of-the-art SSL methods. Moreover, it is foreseeable that as the clients become more label-skewed for Non-IID data partition, the performance of our method degrades. However, even the most label-skewed unlabeled clients can improve the performance of the labeled server using our approach. A limitation of our work is that as the supervised data size decreases, the performance of SemiFL degrades more than the centralized SSL methods. We believe it is because we cannot train labeled and unlabeled data simultaneously in one data batch.

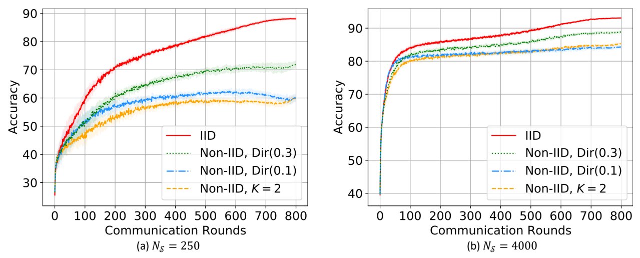

Figure 5. Results of CIFAR10 dataset with (a) $N_{\mathcal{S}} = 250$ and (b) $N_{\mathcal{S}} = 4000$.

Comparison with FL and SSFL Methods We compare our results with the state-of-the-art FL and SSFL methods in Table 1. We demonstrate that SemiFL can perform competitively with the state-of-the-art FL result trained with fully supervised data. It is worth mentioning that SSFL may outperform FL methods in the Non-IID data partition case because the server has a small set of labeled IID data. We also demonstrate that our method significantly outperforms existing SSFL methods. We note that existing SSFL methods fail to perform closely to the state-of-the-art centralized

Table 1. Comparison of SemiFL with the Baseline, SSL, FL, and SSFL methods. SemiFL improves the performance of the labeled server, SemiFL significantly outperforms the existing SSFL methods, and performs close to the state-of-the-art FL and SSL methods.

Dataset			CIFAR10		SVHN		CIFAR100	
Number of Supervised			250	4000	250	1000	2500	10000
Baseline		Fully Supervised	95.3(0.1)		97.3(0.0)		79.3(0.1)	
		Partially Supervised	42.4(1.8)	76.9(0.2)	77.1(2.9)	90.4(0.5)	27.2(0.7)	59.3(0.1)
SSL		Π-Model (Rasmus et al., 2015)	45.7(4.0)	86.0(0.4)	81.0(1.9)	92.5(0.4)	42.8(0.5)	62.1(0.1)
		Pseudo-Labeling (Tarvainen & Valpola, 2017)	50.2(0.4)	83.9(0.3)	79.8(1.1)	90.1(0.6)	42.6(0.5)	63.8(0.2)
		Mean Teacher (Tarvainen & Valpola, 2017)	67.7(2.3)	90.8(0.2)	96.4(0.1)	96.6(0.1)	46.1(0.6)	64.2(0.2)
		MixMatch (Berthelot et al., 2019b)	89.0(0.9)	93.6(0.1)	96.0(0.2)	96.5(0.3)	60.1(0.4)	71.7(0.3)
		UDA (Xie et al., 2019)	91.2(1.1)	95.1(0.2)	94.3(2.8)	97.5(0.2)	66.9(0.2)	75.5(0.3)
		ReMixMatch (Berthelot et al., 2019a)	94.6(0.1)	95.3(0.1)	97.1(0.5)	97.4(0.1)	72.6(0.3)	77.0(0.6)
		FixMatch (Sohn et al., 2020a)	94.9(0.7)	95.7(0.1)	97.5(0.4)	97.7(0.1)	71.7(0.1)	77.4(0.1)
Non-IID, $K = 2$	FL	HeteroFL (Diao et al., 2021)	51.5(3.6)		72.3(4.4)		3.1(0.3)	
	SSFL	FedMatch (Jeong et al., 2020)	32.3(1.9)	61.5(0.7)	61.1(1.6)	52.6(1.4)	18.2(0.3)	31.8(0.6)
		FedRGD (Zhang et al., 2020)	32.7(3.6)	48.9(1.4)	21.2(2.2)	21.6(2.3)	13.8(1.4)	26.5(3.0)
		SemiFL	**60.0(0.9)**	**85.3(0.3)**	**87.5(1.1)**	**92.2(0.8)**	**35.2(0.3)**	**62.1(0.4)**
Non-IID, $\mathrm{Dir}(0.1)$	FL	HeteroFL (Diao et al., 2021)	85.0(0.6)		95.8(0.1)		74.0(0.4)	
	SSFL	FedMatch (Jeong et al., 2020)	32.8(1.6)	63.2(1.3)	60.5(1.2)	53.9(1.1)	18.1(0.1)	32.0(0.9)
		FedRGD (Zhang et al., 2020)	31.5(2.9)	45.2(0.8)	20.0(4.0)	23.8(3.4)	13.4(1.3)	23.6(2.6)
		SemiFL	**63.0(0.6)**	**84.5(0.4)**	**91.2(0.3)**	**93.0(0.5)**	**49.0(1.0)**	**68.0(0.2)**
Non-IID, $\mathrm{Dir}(0.3)$	FL	HeteroFL (Diao et al., 2021)	91.6(0.1)		96.8(0.0)		76.9(0.1)	
	SSFL	FedMatch (Jeong et al., 2020)	31.7(0.9)	62.0(0.7)	61.6(0.8)	53.9(1.2)	18.1(0.1)	32.2(0.4)
		FedRGD (Zhang et al., 2020)	32.5(3.0)	46.9(1.6)	24.8(5.1)	22.0(3.9)	13.1(2.0)	23.8(1.9)
		SemiFL	**71.9(1.2)**	**88.9(0.3)**	**94.0(0.5)**	**95.2(0.2)**	**54.9(1.4)**	**70.0(0.3)**
IID	FL	HeteroFL (Diao et al., 2021)	94.3(0.1)		97.5(0.0)		77.8(0.2)	
	SSFL	FedMatch (Jeong et al., 2020)	32.9(1.4)	59.7(1.5)	63.6(1.3)	56.9(2.2)	18.1(0.3)	31.7(0.6)
		FedRGD (Zhang et al., 2020)	33.2(1.9)	47.8(1.7)	21.3(6.5)	20.7(1.1)	13.3(1.4)	23.8(2.6)
		SemiFL	**88.2(0.3)**	**93.1(0.1)**	**96.8(0.3)**	**96.9(0.1)**	**61.3(1.2)**	**72.1(0.2)**

SSL methods, even if their underlying SSL methods are the same. Moreover, existing SSFL methods cannot outperform the Partially Supervised case, indicating that they deteriorate the performance of the labeled server. To our best knowledge, *the proposed SemiFL is the first SSFL method that actually improves the performance of the labeled server and performs close to the state-of-the-art FL and SSL methods.*

Table 2. Ablation study on each component of alternative training with CIFAR10 dataset. The combination of "Fine-Tuning with Labeled Data" and "Pseudo-Labeling with Received Model" significantly improve the performance.

Method	Fine-Tuning with Labeled Data	Pseudo-Labeling with Received Model	Accuracy	
			Non-IID, $K = 2$	IID
Fully Supervised	N/A		95.33	
Partially Supervised			76.92	
FedAvg + FixMatch	✗	✗	41.01	40.26
SemiFL	✗	✓	48.89	47.03
	✓	✗	80.42	81.70
	✓	✓	**85.34**	**93.10**

Ablation Study We conduct an ablation study on SemiFL and demonstrate the results in Table 2. Based on our extensive experiments, it is evident that "Fine-Tuning with Labeled Data" and "Pseudo-Labeling with Received Model" are the critical components of the proposed 'Alternate Training' method for the success of our method. We also conduct ablation study on static Batch Normalization(sBN), the number of local training epoch, the Mixup data augmentation, and global SGD momentum. The detailed results can be found in the Appendix.

5. Conclusion

In this work, we propose a new communication efficient Federated Learning (FL) framework referred to as SemiFL to address the problem of Semi-Supervised Federated Learning (SSFL). We identify the difficulty of combining the state-of-the-art Semi-Supervised Learning (SSL) methods with communication efficient FL methods. We develop a theoretical analysis of strong data augmentation for SSL, which can be interesting in its own right. We propose to train the labeled server and unlabeled clients alternatively by 'Fine-Tuning with Labeled Data' and 'Pseudo-Labeling with Received Model.' We utilize several training techniques and establish a strong benchmark for SSFL. Extensive experimental studies demonstrate that our communication efficient method can significantly improve the performance of a labeled server with unlabeled clients. Moreover, we show that SemiFL can perform competitively with the state-of-the-art centralized SSL and fully supervised FL methods. Our study provides a practical FL framework that extends the scope of FL applications.

Acknowledgments

This paper is based upon work supported by the Office of Naval Research under grant number N00014-18-1-2244, the Army Research Laboratory and the Army Research Office under grant number W911NF-20-1-0222, and the National Science Foundation under grant number DMS-2134148.

References

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

Ahn, J.-H., Simeone, O., and Kang, J. Wireless federated distillation for distributed edge learning with heterogeneous data. In *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6. IEEE, 2019.

Albaseer, A., Ciftler, B. S., Abdallah, M., and Al-Fuqaha, A. Exploiting unlabeled data in smart cities using federated learning. *arXiv preprint arXiv:2001.04030*, 2020.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Audibert, J.-Y. and Tsybakov, A. B. Fast learning rates for plug-in classifiers under the margin condition. *arXiv preprint math/0507180*, 2005.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Bachman, P., Alsharif, O., and Precup, D. Learning with pseudo-ensembles. *arXiv preprint arXiv:1412.4864*, 2014.

Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019b.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Dai, Z., Yang, Z., Yang, F., Cohen, W. W., and Salakhutdinov, R. Good semi-supervised learning that requires a bad gan. *arXiv preprint arXiv:1705.09783*, 2017.

Devroye, L., Györfi, L., and Lugosi, G. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

Diao, E., Ding, J., and Tarokh, V. HeteroFL: Computation and communication efficient federated learning for heterogeneous clients. In *International Conference on Learning Representations*, 2021.

French, G., Mackiewicz, M., and Fisher, M. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

Grandvalet, Y., Bengio, Y., et al. Semi-supervised learning by entropy minimization. In *CAP*, pp. 281–296, 2005.

Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. *A distribution-free theory of nonparametric regression*, volume 1. Springer, 2002.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

He, C., Li, S., So, J., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M., and Avestimehr, S. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Itahara, S., Nishio, T., Koda, Y., Morikura, M., and Yamamoto, K. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *arXiv preprint arXiv:2008.06180*, 2020.

Ivkin, N., Rothchild, D., Ullah, E., Stoica, I., Arora, R., et al. Communication-efficient distributed sgd with sketching. In *Advances in Neural Information Processing Systems*, pp. 13144–13154, 2019.

Jeong, W., Yoon, J., Yang, E., and Hwang, S. J. Federated semi-supervised learning with inter-client consistency & disjoint learning. *arXiv preprint arXiv:2006.12097*, 2020.

Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Jin, Y., Wei, X., Liu, Y., and Yang, Q. Towards utilizing unlabeled data in federated learning: A survey and prospective. *arXiv e-prints*, pp. arXiv–2002, 2020.

Khodak, M., Balcan, M.-F. F., and Talwalkar, A. S. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, pp. 5917–5928, 2019.

Kohler, M. and Krzyzak, A. On the rate of convergence of local averaging plug-in classification rules under a margin condition. *IEEE transactions on information theory*, 53 (5):1735–1742, 2007.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

Li, A., Sun, J., Wang, B., Duan, L., Li, S., Chen, Y., and Li, H. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020a.

Li, D. and Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020b.

Li, X., Jiang, M., Zhang, X., Kamp, M., and Dou, Q. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

Liang, P. P., Liu, T., Ziyin, L., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

Long, Z., Che, L., Wang, Y., Ye, M., Luo, J., Wu, J., Xiao, H., and Ma, F. Fedsemi: An adaptive federated semi-supervised learning framework. *arXiv preprint arXiv:2012.03292*, 2020.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McClosky, D., Charniak, E., and Johnson, M. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pp. 152–159, 2006.

McLachlan, G. J. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. 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.

Nadaraya, E. A. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Nishio, T. and Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2019.

Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*, 2015.

Rosenberg, C., Hebert, M., and Schneiderman, H. Semi-supervised self-training of object detection models. 2005.

Sahu, A. K., Li, T., Sanjabi, M., Zaheer, M., Talwalkar, A., and Smith, V. Federated optimization for heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 1(2):3, 2018.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *arXiv preprint arXiv:1606.04586*, 2016.

Sattler, F., Marban, A., Rischke, R., and Samek, W. Communication-efficient federated distillation. *arXiv preprint arXiv:2012.00632*, 2020.

Scudder, H. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 4424–4434, 2017.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020a.

Sohn, K., Zhang, Z., Li, C.-L., Zhang, H., Lee, C.-Y., and Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020b.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643*, 2019a.

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019b.

Watson, G. S. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Xian, X., Wang, X., Ding, J., and Ghanadan, R. Assisted learning: A framework for multi-organization learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Yang, D., Xu, Z., Li, W., Myronenko, A., Roth, H. R., Harmon, S., Xu, S., Turkbey, B., Turkbey, E., Wang, X., et al. Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan. *Medical image analysis*, 70:101992, 2021.

Yoon, T., Shin, S., Hwang, S. J., and Yang, E. Fedmix: Approximation of mixup under mean augmented federated learning. In *International Conference on Learning Representations*, 2021.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, Z., Yang, Y., Yao, Z., Yan, Y., Gonzalez, J. E., and Mahoney, M. W. Improving semi-supervised federated learning by reducing the gradient diversity of models. *arXiv preprint arXiv:2008.11364*, 2020.

Zhao, Y., Liu, H., Li, H., Barnaghi, P., and Haddadi, H. Semi-supervised federated learning for activity recognition. *arXiv preprint arXiv:2011.00851*, 2020.

Zhou, Z.-H. and Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.

Zou, Y., Yu, Z., Kumar, B., and Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 289–305, 2018.

Appendix for SemiFL

The **Appendix** contains additional discussions, further experimental results, and technical results.

A. Performance Goal

We outline the general performance goal of Semi-Supervised Federated Learning. The performance ceiling is obviously that of the Fully Supervised Learning (FSL) (namely, assuming that all the server's and clients' data are centralized and fully labeled). For our context where clients' data are unlabeled, a vanilla approach trains the labeled data only at the server-side, referred to as Partially Supervised Learning (PSL). Clearly, the PSL performance can serve as a lower bound benchmark for other approaches that employ additional unlabeled data. When the server contains a small amount of labeled data and a substantial amount of unlabeled data (centralized), the Semi-Supervised Learning (SSL) seeks the use of unlabeled data to improve over the PSL. It was shown that state-of-the-art SSL methods such as FixMatch (Sohn et al., 2020a) could produce similar results as FSL.

Our work focuses on Semi-Supervised Federated Learning (SSFL), where the unlabeled data are distributed among many clients. The general goal of SSFL is to perform similarly to the state-of-the-art SSL, and significantly outperform PSL and the existing SSFL methods. In other words, our performance goal is to achieve FSL $\gtrsim$ SSL $\gtrsim$ SSFL $\gg$ PSL.

B. Static Batch Normalization

We utilize a recently proposed adaptation of Batch Normalization (BN) named Static Batch Normalization (sBN) (Diao et al., 2021). It was shown that this method greatly accelerates the convergence and improves the performance of FedAvg (McMahan et al., 2017) compared with other forms of normalization, including InstanceNorm (Ulyanov et al., 2016), GroupNorm (GN) (Wu & He, 2018), and LayerNorm (Ba et al., 2016). During the training phase, sBN does not track the running statistics with momentum as in BN. Instead, it simply standardizes the data batch x_b and utilizes batch-wise statistics μ_b and σ_b in the following way.

$$\tilde{x}_b = \frac{x_b - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}} \cdot \gamma + \beta, \quad \mu_b = \mathrm{E}[x_b], \quad \sigma_b^2 = \mathrm{Var}[x_b]$$

In FL training, the affine parameters γ and β can be aggregated as usual. We note that FedAvg with vanilla BN is not functional because the BN statistics μ and σ used for inference is averaged from the tracked running BN statistics of local clients during training. Let x_m represents the local data of client m (with size N_m). For a total of M local clients, sBN computes the global BN statistics μ and σ for inference by querying each local client one more time after training is finished, based on

$$\mu = \frac{\sum_{m=1}^{M} N_m \mu_m}{\sum_{m=1}^{M} N_m}, \ \mu_m = \mathrm{E}[x_m], \ \sigma_m^2 = \mathrm{Var}[x_m],$$
$$\sigma^2 = \frac{\sum_{m=1}^{M} \left[(N_m - 1)\sigma_m^2 + N_m(\mu_m - \mu)^2\right]}{(\sum_{m=1}^{M} N_m) - 1}.$$

In the context of SemiFL, we need to generate pseudo-labels at every communication round. Thus, local clients need to upload BN statistics for every communication round. Fortunately, we can utilize the server data x_s to update the global statistics instead of querying each local client, where $\mu = \mathrm{E}[x_s]$ and $\sigma^2 = \mathrm{Var}[x_s]$. We provide experimental results of querying the sBN statistics from all the clients and include an ablation study using only the server data in Table 3. In Table 3, we demonstrate the ablation study of the sBN statistics on the CIFAR10 dataset. Compared with updating the sBN statistics with only the server data, updating the sBN statistics with both server and clients does not provide significant improvements.

Table 3. Ablation study of sBN statistics for the CIFAR10 dataset. The alternative way of using the server data to update the global sBN statistics does not degrade the performance.

sBN statistics	250		4000	
	Non-IID, $K = 2$	IID	Non-IID, $K = 2$	IID
server only	60.0(0.8)	86.3(0.2)	85.5(0.1)	93.1(0.2)
server and clients	60.0(0.9)	85.3(0.3)	88.2(0.3)	93.1(0.1)

C. Further Experimental Results

C.1. Experimental Setup

In Table 4, we provide the hyperparameters used in the experiments. Similar to (Sohn et al., 2020a), we use SGD as our optimizer and a cosine learning rate decay as our scheduler (Loshchilov & Hutter, 2016). We also use the same hyperparameters as (Sohn et al., 2020a), where the local learning rate $\eta = 0.03$, the local momentum $\beta_l = 0.9$, and the confidence threshold $\tau = 0.95$. The Mixup hyperparameter a is set to be 0.75 as suggested by (Zhang et al., 2017).

We use the standard supervised loss to train the labeled server. For training the unlabeled clients, the "fix" loss L_{fix} (proposed in FixMatch (Sohn et al., 2020a)) leverages the techniques of consistency regularization and pseudo-labeling simultaneously. Specifically, the pseudo-labels are generated from weakly augmented data, and the model is trained with strongly augmented data. The "mix" loss (adapted from MixMatch (Zhang et al., 2017; Berthelot et al., 2019b)) reduces the memorization of corrupted labels and increases the robustness to adversarial examples. It was also shown to benefit the SSL (Berthelot et al., 2019a) and FL (Yoon et al., 2021) methods. We have conducted an ablation study and demonstrated that the mix loss moderately improves performance.

Table 4. Hyperparameters used in our experiments.

	Dataset	CIFAR10		SVHN		CIFAR100	
	Number of Supervised	250	4000	250	1000	2500	10000
	Architecture	WResNet28x2				WResNet28x8	
Server	Batch size	10	250	10	250	10	250
	Epoch	5					
	Optimizer	SGD					
	Learning rate	3.0E-02					
	Weight decay	5.0E-04					
	Momentum	0.9					
	Nesterov	✓					
Client	Batch size	10					
	Epoch	5					
	Optimizer	SGD					
	Learning rate	3.0E-02					
	Weight decay	5.0E-04					
	Momentum	0.9					
	Nesterov	✓					
Global	Communication round	800					
	Momentum	0.5					
	Scheduler	Cosine Annealing					

C.2. Results

In Figure 6 and 8, we demonstrate the results of SVHN and CIFAR100 datasets.

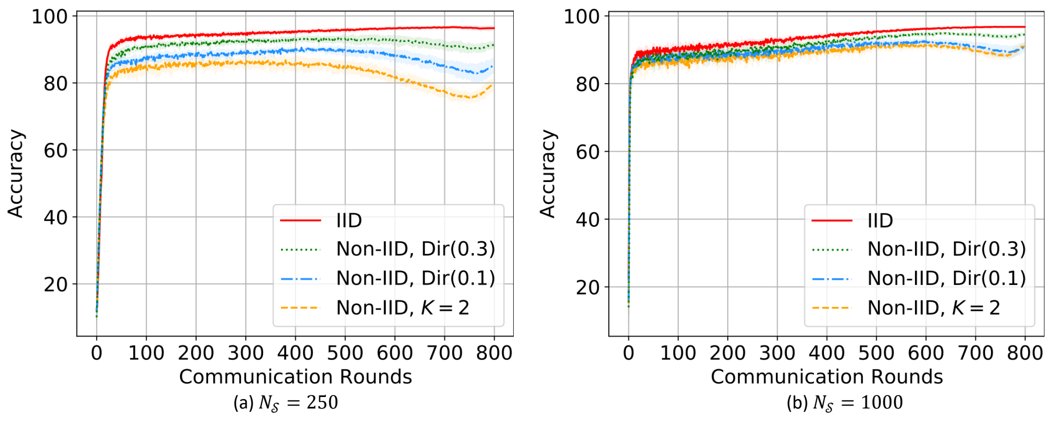

Figure 6. Experimental results for SVHN dataset with (a) $N_{\mathcal{S}} = 250$ and (b) $N_{\mathcal{S}} = 1000$.

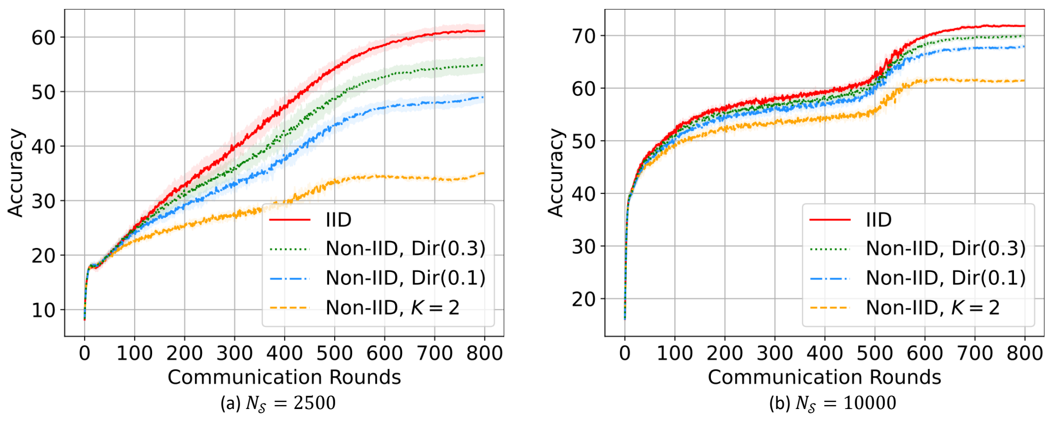

Figure 7. Experimental results for CIFAR100 dataset with (a) $N_{\mathcal{S}} = 2500$ and (b) $N_{\mathcal{S}} = 10000$.

C.3. Ablation Studies

We perform ablation studies of the training techniques adopted in our experiments. We study the efficacy of the number of local training epoch E, the Mixup data augmentation, and the global SGD momentum β_g (Wang et al., 2019a) as shown in Table 5. Less local training epoch significantly hurts the performance due to slow convergence. The Mixup data augmentation has around 2% Accuracy improvement for CIFAR10 dataset. It demonstrates that it is beneficial to combine strong data augmentation with Mixup data augmentation for training unlabeled data. The global momentum marginally improves the result.

Table 5. Ablation study on the CIFAR10 datasets with 4000 labeled data at the server.

E	β_g	Mixup	SemiFL	
			Non-IID, $K = 2$	IID
1	0.5	✓	83.4(0.5)	88.9(0.3)
5	0.5	✗	84.2(0.4)	91.3(0.2)
5	0	✓	85.4(0.6)	92.4(0.1)
5	0.5	✓	85.3(0.3)	93.1(0.1)

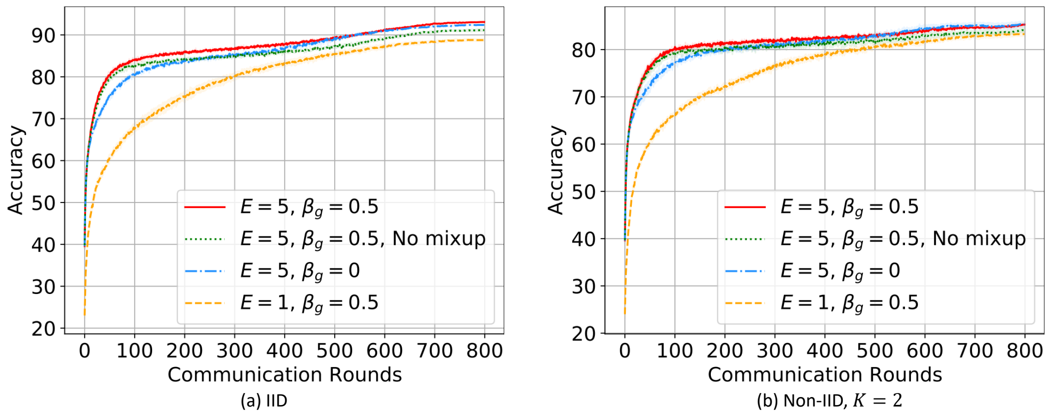

Figure 8. Ablation study of the CIFAR10 dataset with 4000 labeled data at the server for the cases of (a) IID and (b) Non-IID, $K = 2$ data partition.

D. Theoretical Analysis of Strong Data Augmentation for SSL

D.1. Background of Classification

We take the binary classification task as an illustrating example. Let (Y, X) be a random variable with values in $\mathbb{R}^d \times \{1, 0\}$. For the prediction task, we look for a classifier $C : \mathbb{R}^d \to \{1, 0\}$ such that the risk $\mathbb{P}(C(X) \neq Y)$ is small, where $\mathbb{P}$ denotes the probability measure for (Y, X). Let $m(x) \triangleq \mathbb{E}(Y = 1 \mid X = x)$ denote the conditional probability of Y given $X = x$. For example, the standard logistic regression model is in the form of $m(x) = 1/(1 + \exp(-\beta^{\mathrm{T}} x))$ for some $\beta \in \mathbb{R}^d$.

When the underlying m is known, the risk-optimal classifier is known to be

$$C : x \mapsto \mathbb{1}\{m(x) - 1/2\} \tag{7}$$

for any given x. When the underlying m is unknown, we need to train a classifier $\hat{C}_n$ from observed training data (Y_i, X_i), $i = 1, \ldots, n$, which are often assumed to be IID random variables following the same distribution of (Y, X). A general approach is to first learn $\hat{m}_n : \mathbb{R}^d \to \mathbb{R}$ and then let $\hat{C}_n(x) \triangleq \mathbb{1}\{\hat{m}_n(x) - 1/2\}$. To evaluate the prediction performance of a learned $\hat{C}_n$, we consider its gap with the optimal classifer

$$\mathcal{R}(\hat{C}_n) \triangleq \mathbb{P}(Y \neq \hat{C}_n(X)) - \mathbb{P}(Y \neq C(X)) \tag{8}$$

referred to as the classification risk of $\hat{C}_n$.

D.2. Background of Semi-Supervised Learning

Suppose that we observe n_{l} IID labeled data of $(Y^{\mathrm{l}}, X^{\mathrm{l}})$, denoted by $D^{\mathrm{l}} = \{(Y_i^{\mathrm{l}}, X_i^{\mathrm{l}})\}_{i=1}^{n_{\mathrm{l}}}$, where X^{l} has probability distribution $\mathbb{P}_{\mathrm{l}}$ and $\mathbb{E}(Y^{\mathrm{l}} \mid X^{\mathrm{l}} = x) = m(x)$. We also observe n_{u} unlabeled data of (X^{u}), denoted by $\{X_j^{\mathrm{u}}\}_{j=1}^{n_{\mathrm{u}}}$, where each X^{u} has probability distribution $\mathbb{P}_{\mathrm{u}}$. Here, $\mathbb{P}_{\mathrm{u}}$ may or may not be the same as $\mathbb{P}_{\mathrm{l}}$. The Semi-Supervised Learning problem of interest concerns the case $n_{\mathrm{u}} \gg n_{\mathrm{l}}$ and solutions that can properly utilize the unlabeled data to boost the performance of a classifier trained from labeled data. In other words, we look for a classifier $\hat{C}_n^{\mathrm{ssl}}(x)$ trained from observations of both $(Y^{\mathrm{l}}, X^{\mathrm{l}})$ and X^{u}, so that its risk satisfies

$$\mathcal{R}(\hat{C}^{\mathrm{ssl}}) \ll \mathcal{R}(\hat{C}^{\mathrm{l}})$$

where $\hat{C}^{\mathrm{l}}$ is the classifier trained from observations of $(Y^{\mathrm{l}}, X^{\mathrm{l}})$ only.

D.3. A new perspective of Semi-Supervised Learning

As we mentioned in Section 2, there has been a lot of empirical success in using new techniques such as consistency regularization and strong augmentation to improve the classification risk of classical Semi-Supervised Learning. Recently,

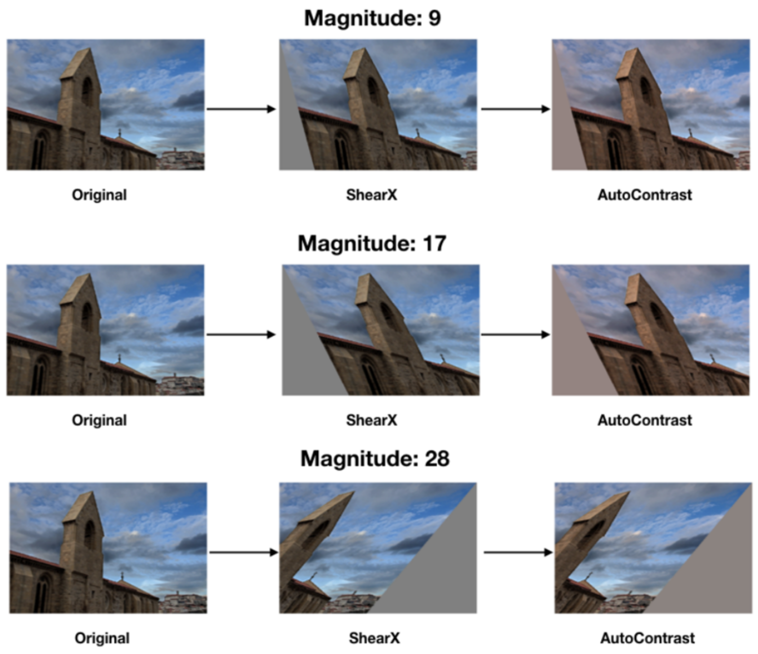

Figure 9. Example of strong data augmentations based on the RandAugment technique (Cubuk et al., 2020). As the distortion magnitude increases, the strength of the augmentation increases. Here, "ShearX" means shearing the image along the horizontal axis, and "AutoConstrast" means maximizing the image contrast by setting the darkest (respectively lightest) pixel to black (respectively white).

the work of (Wei et al., 2020) provides a theoretical understanding of the consistency regularization in reducing classification risk. Its analysis is based on an "expansion" assumption that a low-probability subset of data must expand to a large-probability neighborhood, and there is little overlap between neighborhoods of different classes. To the best of our knowledge, the existing theories do not explain why the strong augmentation technique works so well (to achieve state-of-the-art performance) for Semi-Supervised Learning. Intuitively, strong augmentation is a process that maps a data point (e.g., an image) from high quality to relatively low quality in a unilateral manner (illustrated in Figure 9). Strong augmentation such as RandAugment (Cubuk et al., 2020) consists of a set of data augmentation strategies, e.g., rotating the image, shearing the image, translating the image, adjusting the color balance, and modifying the brightness. The low-quality data and their high-confidence pseudo-labels are then used for training so that there are sufficient "observations" near the difficult data regimes (e.g., near the decision boundary).

In line with the above intuition, we develop a theoretical understanding of how and when using strong augmentation can significantly reduce the classification risk obtained from only labeled data. Instead of studying Semi-Supervised Learning in full generality, we restrict our attention to a class of nonparametric kernel-based classification learning and derive analytically tractable statistical risk-rate analysis. Our theory is based on an intuitive "*adequate transmission*" assumption, which basically means that the distribution of augmented data from high-confidence unlabeled data can adequately cover the data regime of interest during the test. Consequently, reliable information exhibited from unlabeled data can be "transmitted" to data regimes that may have been insufficiently trained with labeled data.

In addition to the notations made in Subsections D.1 and D.2, we will let $\tilde{X}$ denote strongly-augmented data from X^{u}, and $\tilde{Y}$ its corresponding label that follows the same conditional distribution, namely $\mathbb{P}(\tilde{Y}=1 \mid \tilde{X}) = m(\tilde{X})$. Recall that $\mathbb{P}_{\mathrm{u}}$ and $\mathbb{P}_{\mathrm{l}}$ are the probability measures of unlabeled X^{u} and labeled X^{l}, respectively. We suppose that the test data distribution for evaluating the classification performance also follows $\mathbb{P}_{\mathrm{l}}$. In other words, the probability measure in (8) is the product of $\mathbb{P}_{Y|X}$ or $\mathbb{P}_{\tilde{Y}|\tilde{X}}$ (as determined by $m(\cdot)$) and $\mathbb{P}_{\mathrm{l}}$. Let $\hat{m}_0$ denote an initial estimate of m. For generality, we will assume $\hat{m}_0$ is learned from all or only part of the available labeled data. To develop theoretical analyses, we consider the following generic SSL classifier with strong augmentation.

Generic Semi-Supervised Learning with Strong Data Augmentation

• *Step 1*. From $\{X_i^{\mathrm{u}}\}_{i=1}^{n_{\mathrm{u}}}$, we pick up those "high-confidence" x satisfying

$$\min\{1-\hat{m}_0(x), \hat{m}_0(x)\} \leq \delta \tag{9}$$

for some δ (to be quantified), and denote the set as $\mathcal{X}^{\mathrm{aug}}$.
• *Step 2*. For each $X \in \mathcal{X}^{\mathrm{aug}}$, we calculate the pseudo-label $\hat{Y} = \mathbb{1}\{\hat{m}_0(X) - 1/2\}$; meanwhile, we generate the strongly augmented data $\tilde{X}$. Consequently, we obtain a set of data $(\hat{Y}, \tilde{X})$ and denote that set as D^{aug}.
• *Step 3*. Train an estimate of m, denoted by m^{ssl}, and the associated classifier C^{ssl} using the labeled and augmented data $D^{\mathrm{ssl}} \triangleq D^{\mathrm{l}} \cup D^{\mathrm{aug}}$.

Note that if $\hat{m}_0$ is learned from data independent with D^{l}, the data in D^{ssl} are independent but not necessarily identically distributed (since $\mathbb{P}_{\mathrm{l}}$ and $\mathbb{P}_{\mathrm{u}}$ may not be the same).

To show how SSL with strong augmentation can potentially enhance classification learning, we consider a classical nonparametric classifier $\hat{C}$ defined in the following way. Let $K : \mathbb{R}^d \to \mathbb{R}^+$ denote the box kernel function that maps u to $\mathbb{1}\{\|u\| \leq 1\}$, where $\mathbb{1}\{\cdot\}$ denotes the indicator function. With n labeled data (Y_i, X_i), similarly to (7), we define

$$\hat{C}_n : x \mapsto \mathbb{1}\{\hat{m}_n(x) - 1/2\}, \quad \text{where } \hat{m}_n(x) = \frac{\sum_{i=1}^n K(h_n^{-1}(x - X_i)) \cdot Y_i}{\sum_{i=1}^n K(h_n^{-1}(x - X_i))} \tag{10}$$

if $\sum_{i=1}^n K(h_n^{-1}(x - X_i)) \neq 0$, and $\hat{m}_n(x) = 0$ otherwise. Here, $\hat{m}_n$ is known as the Nadaraya-Watson kernel estimate (Nadaraya, 1964; Watson, 1964) of the underlying m, and $h_n > 0$ is the bandwidth.

In our setting, we suppose that $n_0 > 0$ labeled data are used to learn $\hat{m}_0$, and another $n_{\mathrm{l}} \geq 0$ labeled data along with $n_{\mathrm{u}} > 0$ unlabeled data to learn $\hat{m}^{\mathrm{ssl}}$ and thus the subsequent classifier $\hat{C}^{\mathrm{ssl}}$. Note that the n_{l} is introduced only for generality. Our technical analysis includes $n_{\mathrm{l}} = 0$ as a special case. In the main result to be introduced, the risk bound will only involve n_{u} but eliminate n_{l} during technical derivations since we are interested in the regime of $n_{\mathrm{u}} \gg n_0 + n_{\mathrm{l}}$.

Before starting the main result, we make the following additional technical assumptions and provide the intuitions.

(A1) There exists positive constants c_1 and s such that $\mathbb{P}_{\mathrm{u}}(\min\{1 - m(X), m(X)\} \leq \delta) \geq g_s(\delta)$ for all sufficiently small $\delta > 0$, where $g_s(\delta) \triangleq c_1\delta^s$.

Explanation of (A1): Recall that $\mathbb{P}_{\mathrm{u}}$ is the probability measure of unlabeled data. This condition requires a nontrivial amount of unlabeled data with high confidence (or large margin) in the sense that $m(X)$ is close to either zero or one. The function g_s quantifies the "sufficiency" of data at the tail part of X. Take logistic regression $m(x) = 1/(1 + \exp(-\beta^{\mathrm{T}}x))$ as an example. It can be easily verified that

$$\mathbb{P}_{\mathrm{u}}(1 - m(X) \leq \delta) \geq \mathbb{P}_{\mathrm{u}}(\beta^{\mathrm{T}}X \geq -\log\delta), \quad \mathbb{P}_{\mathrm{u}}(m(X) \leq \delta) \geq \mathbb{P}_{\mathrm{u}}(\beta^{\mathrm{T}}X \leq \log\delta),$$

so $\mathbb{P}_{\mathrm{u}}(\min\{1 - m(X), m(X)\} \leq \delta) = \mathbb{P}_{\mathrm{u}}(1 - m(X) \leq \delta) + \mathbb{P}_{\mathrm{u}}(m(X) \leq \delta) \geq \mathbb{P}_{\mathrm{u}}(|\beta^{\mathrm{T}}X| \geq -\log\delta)$ for all $\delta \in (0, 1/2)$. For example, if $|\beta^{\mathrm{T}}X|$ follows standard Exponential, we let $g_s : \delta \mapsto \delta$.

(A2) There exists a constant $c_3 \in (0, 1/2)$ such that the strong augmentation $X^{\mathrm{u}} \to \tilde{X}$ satisfies $\mathbb{P}(\tilde{Y} = 1 \mid \tilde{X} = \tilde{x}, X^{\mathrm{u}} = x) = m(x)$ for all x such that $\min\{1 - \hat{m}_0(x), \hat{m}_0(x)\} \leq c_3$.

Explanation of (A2): Let us think X^{u} as a high-confidence image, with $m(X^{\mathrm{u}})$ close to either zero or one. Meanwhile, $\tilde{X}$ is a strongly augmented version of X^{u}, e.g., by random masking or noise injection, so $m(\tilde{X})$ is closer to $1/2$ than $m(X^{\mathrm{u}})$. The condition of (A2) means that if conditioning on both images, the label $\tilde{Y}$ has a distribution that is only determined by the high-quality image, which is quite intuitive. A mathematically equivalent way to describe (A2) is that $\tilde{X} \to X^{\mathrm{u}} \to \tilde{Y}$ follows a Markov chain.

(A3) There exist positive constants c_2, c_4, and a non-negative v such that for every $\mathbb{P}_{\mathrm{l}}$-measurable ball $B \subseteq \mathbb{R}^d$ with $\mathbb{P}_{\mathrm{l}}(B) \leq c_4$, for the strong augmentation $X^{\mathrm{u}} \to \tilde{X}$, we have $\mathbb{P}_{\mathrm{u}}(\tilde{X} \in B \mid \min\{1 - \hat{m}_0(X^{\mathrm{u}}), \hat{m}_0(X^{\mathrm{u}})\} \leq \delta)/\mathbb{P}_{\mathrm{l}}(B) \geq g_v(\delta)$ for all sufficiently small $\delta > 0$, where $g_v(\delta) \triangleq c_2\delta^v$.

Explanation of (A3): The above numerator is the probability of the augmented data $\tilde{X}$ falling into B conditional on the original unlabeled data (with probability $\mathbb{P}_{\mathrm{u}}$) has high confidence. This assumption ensures that for every regime of significant interest in evaluating the prediction performance (since $\mathbb{P}_1$ is the measure for test data), there will be a sufficient probability coverage of the augmented data. This is an intuitive condition since otherwise, the augmented data cannot represent the test data of interest to boost the test performance. In this assumption, the function g_v determines the coverage as a function of tail probability δ. For example, if $v = 0$, a sufficiently small δ (or higher confidence) gives a non-vanishing coverage. The combination of (A2) and (A3) can be interpreted as an "adequate transmission" condition, under which a small amount of high-confidence unlabeled data can induce augmented data that can accurately represent the test data regime of interest. Such transmitted data can be basically approximated as labeled data for supervised training.

(A4) There exist positive constants c_6 and α such that $\mathbb{P}_1(|m(X^1) - 1/2| \leq t) \leq c_6 t^{\alpha}$ for all $t > 0$. Moreover, $X^1 \in [0, 1]^d$.

Explanation of (A4): The inequality is a margin condition that has been used in the classical learning literature (see, e.g., (Devroye et al., 2013; Kohler & Krzyzak, 2007) and the references therein). It determines the difficulty of the underlying classification task. Intuitively speaking, a larger α means more separability of the two classes under the probability $\mathbb{P}_1$. The boundedness of X^1 is for technical convenience.

(A5) There exist positive constants q and c_7 such that $|m(x) - m(x')| \leq c_7 \|x - x'\|^q$ for all $x, x' \in [0, 1]^d$, where $\| \cdot \|$ denotes the Euclidean norm.

Explanation of (A5): This condition assumes a Lipschitz-type condition of $m(\cdot)$, where q is allowed to be different from one. Intuitively, it assumes the underlying classifier to learn cannot be too bumpy. For $q \in (0, 1]$, a larger q means more smoothness of $m(\cdot)$.

(A6) There exist positive constants r, c_8, and Δ such that $|\hat{m}_0(x) - m(x)| \leq c_8 n_0^{-r}$ for all x satisfying $\min\{1 - \hat{m}_0(x), \hat{m}_0(x)\} \leq \Delta$.

Explanation of (A6): This assumption requires that conditional on X falls into a large-margin area, the estimation error of the initial function $\hat{m}_0$ is not too large.

(A7) For the constants s, v, α, q, and r defined in the above assumptions, we have

$$\frac{q \cdot s}{q \cdot (\alpha + 3 + v + s) + d} < \frac{1}{2}, \tag{11}$$

$$\frac{n_0^{-r}}{n_{\mathrm{u}}^{-q/\{q(\alpha+3+v+s)+d\}}} \to 0, \text{ as } \min\{n_0, n_{\mathrm{u}}\} \to \infty. \tag{12}$$

Explanation of (A7): The two inequalities will be technical conditions used in the proof. A sufficient condition for (11) to hold is that $\alpha \geq s$. Intuitively, this requires that α, which describes the separability of the decision boundary (the larger, the better), is not smaller than s, which quantifies the sufficiency of tail samples (the smaller, the better). The inequality (12) means that the initial classifier $\hat{m}_0$ cannot perform too poorly. This matches our empirical observations that the SSL training in each round has to immediately follow a preceding round that uses some labeled data. Also, the denominator in (12) favors relatively small s, d compared with α, v, q.

D.4. Main result

Our **main result** is provided below.

Theorem 1: Under Assumptions (A1)-(A7), the generic SSL classifier with strong augmentation (namely the above Steps 1-3) satisfies

$$\mathcal{R}(\hat{C}^{\mathrm{ssl}}) \leq C n_{\mathrm{u}}^{-q(\alpha+1)/\{q(\alpha+3+v+s)+d\}} \tag{13}$$

for some constant C that does not depend on the sample size.

Explanation of Theorem 1: The theorem gives an explicit rate of convergence for the SSL classification risk using unlabeled data of size n_{u}. It is the informal statement made in the main paper with $\rho \overset{\Delta}{=} v + s$. We interpret the power

$$\frac{q(\alpha + 1)}{q(\alpha + 3 + v + s) + d}$$

as follows. If the margin parameter α is large, the classification is relatively easy, and the ratio can go up to one, namely $\mathcal{R}(\hat{C}^{\mathrm{ssl}}) \sim n_{\mathrm{u}}^{-1}$. This is reminiscent of an existing result that uses labeled data and large margin to achieve the n_{l}^{-1}

rate (Audibert & Tsybakov, 2005). If the tail sufficiency parameter s or the coverage parameter v is large, the ratio becomes approximately $(\alpha+1)/(v+s)$. Intuitively, a larger s or v indicates that there will be fewer high-confidence unlabeled data to be transmitted to benefit the classification learning (on the evaluation measure $\mathbb{P}_{\text{l}}$ of interest), which is inline with a slower rate of convergence $n_{\text{u}}^{-(\alpha+1)/(v+s)}$.

On the contrary, consider the other extreme that $v = s = 0$. Then, the ratio becomes $q(\alpha+1)/\{q(\alpha+3)+d\}$, which matches an existing result in classification learning (Kohler & Krzyzak, 2007). For comparison, we define the baseline classifier that only uses n_{l} labeled data, based on the kernel estimation in (10). We denote that classifer as $\hat{C}^{\text{l}}$. The risk would be $\mathcal{R}(\hat{C}^{\text{l}}) \leq C' n_{\text{l}}^{-q(\alpha+1)/\{q(\alpha+3)+d\}}$ for some constant C'. Comparing this with (13), we can determine the region where employing SSL can significantly improve supervised learning. To illustrate this point, let us suppose that

$$n_{\text{l}} \sim n_{\text{u}}^{\zeta}$$

for some constant $\zeta \in (0,1)$. It can be verified that the bound of $\mathcal{R}(\hat{C}^{\text{l}})$ is much larger than that of $\mathcal{R}(\hat{C}^{\text{ssl}})$ when

$$\frac{q(\alpha+1)}{q(\alpha+3+v+s)+d} > \frac{\zeta q(\alpha+1)}{q(\alpha+3)+d},$$

or equivalently,

$$\zeta < \frac{q(\alpha+3)+d}{q(\alpha+3+v+s)+d}. \tag{14}$$

The inequality (14) provides an insight into the *critical region of* n_u where significant improvement can be made from unlabeled data, as dependent on constants that describe the underlying function smoothness (q), data dimension (d), task difficulty (α), and "adequate transmission" parameters (s, v).

D.5. Proof of Theorem 1

We first give a sketch of the proof. We first relate the risk bound of $\mathcal{R}(\hat{C}^{\text{ssl}})$ to the estimation error of $\hat{m}^{\text{ssl}}$, and then decompose the error into a bias term and a variance term. Each term is then bounded using concentration inequalities, in a way similar to the techniques used in (Györfi et al., 2002, Ch. 5) and (Kohler & Krzyzak, 2007). Different from the standard nonparametric analysis of classification learning with IID data, we will use the aforementioned "adequate transmission" conditions to derive the rate of convergence from data that are contributed from both labeled and pseudo-labeled data. The analysis involves a careful choice of the tuning parameters, e.g., the δ in Assumption (A1) and the kernel bandwidth, so that the biases introduced from pseudo-labeled data have a diminishing influence on the risk rate. Next, we provide detailed proof.

We let $n = n_{\text{l}} + n_{\text{u}}$ denote the total size of labeled and unlabeled data available to the SSL training. For notational clarity, we sometimes put subscript n, e.g., δ_n instead of δ (in Step 1), to highlight a quantity that is designed to vanish at some rate as n becomes large. Recall that $D^{\text{ssl}} = D^{\text{l}} \cup D^{\text{aug}}$. Let n_{l} and $n_{\text{u}}^{\text{aug}}$ denote the sample sizes of D^{l} and D^{aug}, respectively. Note that $n_{\text{u}}^{\text{aug}}$ is random since the Step 1 depends on n_0 labeled data. We first consider the risk conditional on a fixed $n_{\text{u}}^{\text{aug}}$, denoted by $\mathcal{R}_{n_{\text{u}}^{\text{aug}}}(\hat{C}^{\text{ssl}})$.

Direct calculations show that

$$\begin{aligned}
\mathcal{R}_{n_{\text{u}}^{\text{aug}}}(\hat{C}^{\text{ssl}}) &= \mathbb{E}_{\text{l}}\Big(|2m(X)-1| \cdot \mathbb{1}\{\hat{C}^{\text{ssl}}(X) \neq C(X)\}\Big) = T_1 + T_2, \text{ where} \\
T_1 &= 2\mathbb{E}_{\text{l}}\Big(|m(X)-1/2| \cdot \mathbb{1}\Big\{|m(X)-1/2| \leq t_n, \hat{C}^{\text{ssl}}(X) \neq C(X)\Big\}\Big) \\
T_2 &= 2\mathbb{E}_{\text{l}}\Big(|m(X)-1/2| \cdot \mathbb{1}\Big\{|m(X)-1/2| > t_n, \hat{C}^{\text{ssl}}(X) \neq C(X)\Big\}\Big)
\end{aligned} \tag{15}$$

for an arbitrary $t_n > 0$ to be selected. From Assumption (A4), $|m(X)-1/2| \leq 1/2$, and $\mathbb{1}\{|m(X)-1/2| > t_n, \hat{C}^{\text{ssl}}(X) \neq C(X)\} \leq \mathbb{1}\{|m(X)-\hat{m}(X)| > t_n\}$, we have

$$T_1 \leq 2t_n \cdot \mathbb{P}_{\text{l}}\big(|m(X)-1/2| \leq t_n\big) \leq 2c_6 t_n^{1+\alpha}, \quad T_2 \leq \mathbb{P}_{\text{l}}\big(|m(X)-\hat{m}(X)| > t_n\big). \tag{16}$$

Moreover, by the triangle inequality, we have

$$T_2 \le \mathbb{P}_{\mathrm{l}}\big(|m(X) - \bar{m}(X)| > t_n/2\big) + \mathbb{P}_{\mathrm{l}}\big(|\bar{m}(X) - \hat{m}(X)| > t_n/2\big), \tag{17}$$

where we define the function $\bar{m}$ by

$$\bar{m}(x) = \frac{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))m(X)}{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))}$$

if the denominator is nonzero, and $\bar{m}(x) = 0$ otherwise.

In the sequel, we bound each term in (17). First, we rewrite

$$\mathbb{P}_{\mathrm{l}}\big(|m(X) - \bar{m}(X)| > t_n/2\big) = \int_{x \in [0,1]^d} \mathbb{P}\big(|m(x) - \bar{m}(x)| > t_n/2\big) d\mathbb{P}_{\mathrm{l}}(x), \tag{18}$$

where $\mathbb{P}$ denotes the probability measure induced by D^{ssl} (which is implicitly used to define $\bar{m}$). For each x, we define the event

$$E_x = \bigg\{\omega : \sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))\bigg\}.$$

Then, from Assumption (A5) and the definition that $K(u) = \mathbb{1}\{\|u\| \le 1\}$, we have

$$\begin{aligned}
|m(x) - \bar{m}(x)| &= \frac{|\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))(m(x) - m(X))|}{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))} \cdot \mathbb{1}\{E_x\} + m(x)(1 - \mathbb{1}\{E_x\}) \\
&\le \frac{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))|x-X|^q}{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))} \cdot \mathbb{1}\{E_x\} + m(x)(1 - \mathbb{1}\{E_x\}) \\
&\le c_7 h_n^q + m(x)(1 - \mathbb{1}\{E_x\}).
\end{aligned} \tag{19}$$

Let $B_{x,h} \triangleq \{u \in \mathbb{R}^d : \|u - x\| \le h\}$ denote the Euclidean ball of center x and radius h. If we choose

$$t_n/2 > c_7 h_n^q, \tag{20}$$

the above inequality (19) implies that

$$\begin{aligned}
\mathbb{P}(|m(x) - \bar{m}(x)| \ge t_n/2) &\le \mathbb{P}\bigg(m(x)(1 - \mathbb{1}\{E_x\}) \ge t_n/2 - c_7 h_n^q\bigg) \\
&\le \mathbb{P}\bigg\{\sum_{X \in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X)) = 0\bigg\} \\
&= \mathbb{P}\bigg\{\|x - X\| > h_n, \forall X \in D^{\mathrm{ssl}}\bigg\} \\
&= (1 - \mathbb{P}_{\mathrm{l}}(B_{x,h_n}))^{n_{\mathrm{l}}} \cdot (1 - \mathbb{P}_{\mathrm{u}}(B_{x,h_n}))^{n_{\mathrm{u}}^{\mathrm{aug}}} \qquad (21) \\
&\le \exp\{-n_{\mathrm{l}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})\} \cdot \exp\{-n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{u}}(B_{x,h_n})\} \qquad (22)
\end{aligned}$$

Let $c_9 \triangleq \max_{v>0} ve^v$. Let $\{z_i\}_{i=1}^{M_n}$ be a set of points in $\mathbb{R}^d$ such that $[0,1]^d \subseteq \cup_{i=1}^{M_n} B_{z_i,h_n/2}$, with $M_n = c_{10}h_n^{-d}$ for some

c_{10}. Taking (22) into (18), and invoking Assumption (A3), we obtain

$$\begin{aligned}
&\mathbb{P}_{\mathrm{l}}\big(|m(X) - \bar{m}(X)| > t_n/2\big) \\
&= \int_{x\in[0,1]^d} \exp\{-n_{\mathrm{l}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})\} \cdot \exp\{-n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{u}}(\tilde{X} \in B_{x,h_n} \mid \tilde{X} \in D^{\mathrm{aug}})\} d\mathbb{P}_{\mathrm{l}}(x) \\
&\leq \int_{x\in[0,1]^d} \exp\{-n_{\mathrm{l}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n}) - g_v(\delta_n) n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})\} d\mathbb{P}_{\mathrm{l}}(x) \\
&= \int_{x\in[0,1]^d} \exp\{-\tilde{n}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})\} d\mathbb{P}_{\mathrm{l}}(x) \\
&\leq c_9 \int_{x\in[0,1]^d} \frac{1}{\tilde{n}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})} d\mathbb{P}_{\mathrm{l}}(x) \\
&\leq c_9 \sum_{i=1}^{M_n} \int_{x\in[0,1]^d} \frac{\mathbb{1}\{x \in B_{z_i,h_n/2}\}}{\tilde{n}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})} d\mathbb{P}_{\mathrm{l}}(x) \\
&\leq c_9 \tilde{n}^{-1} M_n = c_9 c_{10} \tilde{n}^{-1} h_n^{-d}
\end{aligned} \tag{23}$$

where we let $\tilde{n} \triangleq n_{\mathrm{l}} + g_v(\delta_n) n_{\mathrm{u}}^{\mathrm{aug}}$. The technique of covering used in the last two inequalities was from (Györfi et al., 2002, Eq. 5.1).

To bound the second term in (17), we write

$$\hat{m}(x) - \bar{m}(x) = \sum_{(Y,X)\in D^{\mathrm{ssl}}} \frac{K(h_n^{-1}(x-X))}{\sum_{(Y,X)\in D^{\mathrm{ssl}}} K(h_n^{-1}(x-X))} (Y - m(X)). \tag{24}$$

Recall that $D^{\mathrm{ssl}} = D^{\mathrm{l}} \cup D^{\mathrm{aug}}$. For every $(Y^{\mathrm{l}}, X^{\mathrm{l}}) \in D^{\mathrm{l}}$, we have $\mathbb{E}(Y^{\mathrm{l}} \mid X^{\mathrm{l}}) = m(X)$.

For any δ_n that satisfies $\delta_n \leq \min\{c_3, \Delta, 1/4\}$, where c_3 was introduced in Assumption (A2) and Δ was introduced in Assumption (A6), we have

$$\begin{aligned}
&\mathbb{P}(\hat{Y} = 1, \tilde{Y} = 0 \mid \tilde{X}, X^{\mathrm{u}}) \\
&= \mathbb{P}(\hat{Y} = 1, \tilde{Y} = 0, \hat{m}_0(X^{\mathrm{u}}) \geq 1 - \delta_n \mid \tilde{X}, X^{\mathrm{u}}) + \mathbb{P}(\hat{Y} = 1, \tilde{Y} = 0, \hat{m}_0(X^{\mathrm{u}}) \leq \delta_n \mid \tilde{X}, X^{\mathrm{u}}) \\
&= \mathbb{P}(\hat{Y} = 1, \tilde{Y} = 0, \hat{m}_0(X^{\mathrm{u}}) \geq 1 - \delta_n \mid \tilde{X}, X^{\mathrm{u}}) \\
&\leq \mathbb{P}(\tilde{Y} = 0, \hat{m}_0(X^{\mathrm{u}}) \geq 1 - \delta_n, m(X^{\mathrm{u}}) \geq 1 - \delta_n - c_8 n_0^{-r} \mid \tilde{X}, X^{\mathrm{u}}) \\
&\quad + \mathbb{P}(\hat{m}_0(X^{\mathrm{u}}) \geq 1 - \delta_n, m(X^{\mathrm{u}}) \leq 1 - \delta_n - c_8 n_0^{-r} \mid \tilde{X}, X^{\mathrm{u}}) \\
&\leq \mathbb{P}(\tilde{Y} = 0, m(X^{\mathrm{u}}) \geq 1 - \delta_n - c_8 n_0^{-r}) + 0 \\
&\leq \delta_n + c_8 n_0^{-r},
\end{aligned}$$

and similarly, $\mathbb{P}(\hat{Y} = 0, \tilde{Y} = 1 \mid \tilde{X}, X^{\mathrm{u}}) \leq \delta_n + c_8 n_0^{-r}$. Thus,

$$\mathbb{E}(|\hat{Y} - \tilde{Y}| \mid \tilde{X}) = \mathbb{E}\{\mathbb{E}(|\hat{Y} - \tilde{Y}| \mid \tilde{X}, X^{\mathrm{u}}) \mid \tilde{X}\} \leq 2\delta_n + 2c_8 n_0^{-r}.$$

Consequently, for every $(\hat{Y}, \tilde{X}) \in D^{\mathrm{aug}}$, we have

$$\mathbb{E}(\hat{Y} \mid \tilde{X}) = \mathbb{E}(\tilde{Y} \mid \tilde{X}) + \kappa(\tilde{X}) = m(\tilde{X}) + \kappa(\tilde{X}) \tag{25}$$

where $\kappa(\tilde{X}) \triangleq \mathbb{E}(\hat{Y} - \tilde{Y} \mid \tilde{X}) \leq 2\delta_n + 2c_8 n_0^{-r}$.

Back in (24), let $u(Y) = Y$ if $(Y, X) \in D^{\mathrm{l}}$ and $u(Y) = \tilde{Y}$ if $(Y, X) \in D^{\mathrm{aug}}$, where $\tilde{Y}$ is the pseudo-label random variable as in Assumption (A2) and equality (25). In this way, we have $\mathbb{E}(u(Y) \mid X) = m(X)$. We rewrite (24) as

$$\begin{aligned}
&\hat{m}(x) - \bar{m}(x) = T_3(x) + T_4(x), \text{ where} \\
&T_3(x) \triangleq \sum_{(Y,X)\in D^{\text{ssl}}} \frac{K(h_n^{-1}(x-X))}{\sum_{(Y,X)\in D^{\text{ssl}}} K(h_n^{-1}(x-X))}(u(Y) - m(X)) \\
&T_4(x) \triangleq \sum_{(\hat{Y},\tilde{X})\in D^{\text{aug}}} \frac{K(h_n^{-1}(x-X))}{\sum_{(Y,X)\in D^{\text{ssl}}} K(h_n^{-1}(x-X))}(\hat{Y} - \tilde{Y}) \\
&\qquad \le \sum_{(\hat{Y},\tilde{X})\in D^{\text{aug}}} \frac{K(h_n^{-1}(x-X))}{\sum_{(\hat{Y},\tilde{X})\in D^{\text{aug}}} K(h_n^{-1}(x-X))}(\hat{Y} - \tilde{Y}).
\end{aligned}$$

Let $\mathcal{X}^{\text{ssl}} \triangleq \{X : (\cdot, X) \in D^{\text{ssl}}\}$ and $\mathcal{X}^{\text{aug}} \triangleq \{X : (\cdot, X) \in D^{\text{aug}}\}$. Then, we can bound

$$\begin{aligned}
&\mathbb{P}\big(|\bar{m}(x) - \hat{m}(x)| > t_n/2 \mid \mathcal{X}^{\text{ssl}}\big) && (26)\\
&\le \mathbb{P}\big(|T_3(x)| > t_n/4 \mid \mathcal{X}^{\text{ssl}}\big) + \mathbb{P}\big(|T_4(x)| > t_n/4 \mid \mathcal{X}^{\text{ssl}}\big) \\
&\le 2\exp\left\{-\frac{2(t_n/4)^2}{\sum_{X\in\mathcal{X}^{\text{ssl}}} K^2(h_n^{-1}(x-X))/\{\sum_{X'} K(h_n^{-1}(x-X'))\}^2}\right\}+ && (27)\\
&\quad + \mathbb{P}\left(\left|\sum_{(\hat{Y},\tilde{X})\in D^{\text{aug}}} \frac{K(h_n^{-1}(x-X))}{\sum_{X'\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X'))}(\hat{Y} - \tilde{Y} - \mathbb{E}(\hat{Y} - \tilde{Y} \mid \tilde{X}))\right| > t_n/8 \mid \mathcal{X}^{\text{ssl}}\right)+ \\
&\quad + \mathbb{P}\left(\left|\sum_{\tilde{X}\in\mathcal{X}^{\text{aug}}} \frac{K(h_n^{-1}(x-\tilde{X}))}{\sum_{X'\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X'))}\kappa(\tilde{X}))\right| > t_n/8 \mid \mathcal{X}^{\text{aug}}\right) \\
&\le 2\exp\left\{-\frac{1}{8}t_n^2 \sum_{X\in\mathcal{X}^{\text{ssl}}} K(h_n^{-1}(x-X))\right\} + 2\exp\left\{-\frac{1}{128}t_n^2 \sum_{X\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X))\right\}+ && (28)\\
&\quad + \mathbb{P}\left(2\delta_n + 2c_8 n_0^{-r} > t_n/8\right) && (29)\\
&\le 4\exp\left\{-\frac{1}{128}t_n^2 \sum_{X\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X))\right\} && (30)\\
&\le 4\mathbb{1}\left\{\sum_{X\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X)) < \frac{1}{2}n_{\text{u}}^{\text{aug}}\mathbb{P}_{\text{u}}(B_{x,h_n}) - \log^2 n_{\text{u}}^{\text{aug}}\right\}+ \\
&\quad 4\exp\left\{-\frac{1}{256}t_n^2 n_{\text{u}}^{\text{aug}}\mathbb{P}_{\text{u}}(B_{x,h_n}) + \frac{1}{128}t_n^2 \log^2 n_{\text{u}}^{\text{aug}}\right\} && (31)
\end{aligned}$$

provided that

$$2\delta_n + 2c_8 n_0^{-r} \le t_n/8. \tag{32}$$

In the above derivation, (27) uses the Hoeffding's inequality, the fact that $K^2(\cdot) = K(\cdot)$, and the triangle inequality, (28) uses the Hoeffding's inequality again, (29) follows from (25), (30) is from $\mathcal{X}^{\text{aug}} \subseteq \mathcal{X}^{\text{ssl}}$, and (31) is by the definition of the indicator function. Consequently, with the choice of

$$t_n \log n_{\text{u}}^{\text{aug}} \le 1, \tag{33}$$

we have

$$\begin{aligned}
&\mathbb{P}\big(|\bar{m}(x) - \hat{m}(x)| > t_n/2\big) && (34)\\
&\le 4\mathbb{P}_{\text{u}}\left\{\sum_{X\in\mathcal{X}^{\text{aug}}} K(h_n^{-1}(x-X)) < \frac{1}{2}n_{\text{u}}^{\text{aug}}\mathbb{P}_{\text{u}}(B_{x,h_n}) - \log^2 n_{\text{u}}^{\text{aug}}\right\} \\
&\quad + 8\exp\left\{-\frac{1}{256}t_n^2 n_{\text{u}}^{\text{aug}}\mathbb{P}_{\text{u}}(B_{x,h_n})\right\}. && (35)
\end{aligned}$$

The first term in (35), according to the Bernstein inequality, can be upper bounded by

$$4\exp\bigg\{-\frac{1}{2}\frac{(n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})/2+\log^2 n_{\mathrm{u}}^{\mathrm{aug}})^2}{n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})+(n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})/2+\log^2 n_{\mathrm{u}}^{\mathrm{aug}})/3}\bigg\}\bigg\}$$
$$\leq 4\exp\bigg\{-\frac{3}{14}(n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{l}}(B_{x,h_n})/2+\log^2 n_{\mathrm{u}}^{\mathrm{aug}})\bigg\}\leq 4\exp\bigg\{-\frac{3}{14}\log^2 n_{\mathrm{u}}^{\mathrm{aug}})\bigg\}.$$

Therefore, we can bound the second term in (17) by

$$\begin{aligned}&\mathbb{P}_{\mathrm{l}}\big(|\bar{m}(X)-\hat{m}(X)|>t_n/2\big)\\ &\leq\int_{x\in[0,1]^d}\mathbb{P}\big(|\bar{m}(x)-\hat{m}(x)|>t_n/2\big)d\mathbb{P}_{\mathrm{l}}(x)\\ &\leq 4\exp\bigg\{-\frac{3}{14}\log^2 n_{\mathrm{u}}^{\mathrm{aug}})\bigg\}+8\int_{x\in[0,1]^d}\exp\bigg\{-\frac{1}{256}t_n^2 n_{\mathrm{u}}^{\mathrm{aug}}\mathbb{P}_{\mathrm{u}}(B_{x,h_n})\bigg\}d\mathbb{P}_{\mathrm{l}}(x).\end{aligned}$$

The second term in (35), according to the same arguments as in (23), can be upper bounded by $8\cdot 256\cdot c_9c_{10}/(g_v(\delta_n)t_n^2 n_{\mathrm{u}}^{\mathrm{aug}}h_n^d)$. Therefore, we have

$$\mathbb{P}_{\mathrm{l}}\big(|\bar{m}(X)-\hat{m}(X)|>t_n/2\big)\leq 4\exp\bigg\{-\frac{3}{14}\log^2 n_{\mathrm{u}}^{\mathrm{aug}})\bigg\}+\frac{2^{11}c_9c_{10}}{g_v(\delta_n)t_n^2 n_{\mathrm{u}}^{\mathrm{aug}}h_n^d}.$$

Combining inequalities (15), (16), (17), and (23), we obtain

$$\mathcal{R}_{n_{\mathrm{u}}^{\mathrm{aug}}}(\hat{C}^{\mathrm{ssl}})\leq 2c_6t_n^{1+\alpha}+\frac{c_9c_{10}}{(n_{\mathrm{l}}+g_v(\delta_n)n_{\mathrm{u}}^{\mathrm{aug}})h_n^d}+4\exp\bigg\{-\frac{3}{14}\log^2 n_{\mathrm{u}}^{\mathrm{aug}})\bigg\}+\frac{2^{11}c_9c_{10}}{g_v(\delta_n)t_n^2 n_{\mathrm{u}}^{\mathrm{aug}}h_n^d}.$$

Finally, we use a probabilistic lower bound of $n_{\mathrm{u}}^{\mathrm{aug}}$ to obtain the risk bound. Let E denote the event $\min\{1-\hat{m}_0(X),\hat{m}_0(X)\}\leq\delta_n$. By the triangle inequality, assumptions (A1) and (A6), we have

$$\begin{aligned}&\mathbb{P}_{\mathrm{u}}(\min\{1-\hat{m}_0(X),\hat{m}_0(X)\}\leq\delta_n)\\ &\geq\mathbb{P}_{\mathrm{u}}(\min\{1-m(X),m(X)\}\leq\delta_n-c_8n_0^{-r})-\mathbb{P}_{\mathrm{u}}(|m(X)-\hat{m}_0(X)|>c_8n_0^{-r},E)\\ &\geq g_s(\delta_n-c_8n_0^{-r})\end{aligned}$$

Note that $n_{\mathrm{u}}^{\mathrm{aug}}$ is a sum of n_{u} IID Bernoulli random variables Z with probability $\mathbb{P}(Z=1)=\mathbb{P}_{\mathrm{u}}(\min\{1-\hat{m}_0(X),\hat{m}_0(X)\}\leq\delta_n)$. By the Hoeffding's inequality, with probability at least $1-2\exp\{-n_{\mathrm{u}}(\tilde{n}_{\mathrm{u}}/n_{\mathrm{u}})^2/2\}$, we have

$$\frac{3\tilde{n}_{\mathrm{u}}}{2}\geq n_{\mathrm{u}}^{\mathrm{aug}}\geq\frac{\tilde{n}_{\mathrm{u}}}{2},\quad\text{where }\tilde{n}_{\mathrm{u}}\overset{\Delta}{=}g_s(\delta_n-c_8n_0^{-r})\cdot n_{\mathrm{u}}.$$

Therefore, we have

$$\begin{aligned}\mathcal{R}(\hat{C}^{\mathrm{ssl}})&=\mathbb{E}\mathcal{R}_{n_{\mathrm{u}}^{\mathrm{aug}}}(\hat{C}^{\mathrm{ssl}})\\ &\leq 2c_6t_n^{1+\alpha}+\frac{c_9c_{10}}{(n_{\mathrm{l}}+g_v(\delta_n)\tilde{n}_{\mathrm{u}}/2)h_n^d}+4\exp\bigg\{-\frac{3}{14}(\log\tilde{n}_{\mathrm{u}}-\log 2)^2)\bigg\}+\\ &\quad\frac{2^{11}c_9c_{10}}{g_v(\delta_n)t_n^2\tilde{n}_{\mathrm{u}}h_n^d/2}+\exp\bigg\{-\frac{n_{\mathrm{u}}}{2}\bigg(g_s(\delta_n-c_8n_0^{-r})\bigg)^2\bigg\},\end{aligned}\tag{36}$$

provided that the choices of (20), (32), and (33) are made, namely

$$t_n/2>c_7h_n^q,\quad 2\delta_n+2c_8n_0^{-r}\leq t_n/8,\quad t_n\log(3\tilde{n}_{\mathrm{u}}/2)\leq 1.$$

Choosing h_n, t_n, and δ_n at the rate of

$$h_n\sim n_{\mathrm{u}}^{-1/\{q(\alpha+3+v+s)+d\}},\quad t_n\sim h_n^q,\quad\delta_n\sim h_n^q,$$

and invoking the assumption (A7), we can verify that the rate of convergence in (36) is at the order of

$$\mathcal{R}(\hat{C}^{\mathrm{ssl}})\sim n_{\mathrm{u}}^{-q(\alpha+1)/\{q(\alpha+3+v+s)+d\}},$$

which concludes the proof.