Private Semi-Supervised Federated Learning

Chenyou Fan[1] , **Junjie Hu**[2] , **Jianwei Huang**[2,3]
[1]School of Artificial Intelligence, South China Normal University, China
[2]Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
[3]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
fanchenyou@scnu.edu.cn, {hujunjie, jianweihuang}@cuhk.edu.cn

Abstract

We study a federated learning (FL) framework to effectively train models from scarce and skewly distributed labeled data. We consider a challenging yet practical scenario: a few data sources own a small amount of labeled data, while the rest mass sources own purely unlabeled data. Classical FL requires each client to have enough labeled data for local training, thus is not applicable in this scenario. In this work, we design an effective federated semi-supervised learning framework (FedSSL) to fully leverage both labeled and unlabeled data sources. We establish a unified data space across all participating agents, so that each agent can generate mixed data samples to boost semi-supervised learning (SSL), while keeping data locality. We further show that FedSSL can integrate differential privacy protection techniques to prevent labeled data leakage at the cost of minimum performance degradation. On SSL tasks with as small as 0.17% and 1% of MNIST and CIFAR-10 datasets as labeled data, respectively, our approach can achieve 5-20% performance boost over the state-of-the-art methods.

1 Introduction

Recently, federated learning (**FL**) [McMahan *et al.*, 2017] has received substantial research interests, as it provides a practical way of training machine learning models with distributed data sources while preserving data privacy. In the FL paradigm, each agent (participant) trains a local learning model with *its owned data only*, while a central server regularly communicates with all agents to generate a better global model through the aggregation of the local models. A key feature of FL is that no direct data exchange happens during the learning process, in contrast to centralized training.

However, existing FL studies assume either each agent owns sufficient training data [McMahan *et al.*, 2017], or the agents collectively have sufficient data for the tasks of interest [Fan and Huang, 2021]. For instance, the image classification task, as the benchmark task of FL [McMahan *et al.*, 2017; Zhao *et al.*, 2018], requires each agent to prepare thousands of *labeled* training samples to fully train the deep-learning based models such as CNNs. In reality, the labeled data is scarce and distributed unevenly over only a few data sources. A typical scenario in mobile computing is that most end users create user data (e.g., tweets), while only a few users have the interests and time to annotate them (e.g., sentiments). The huge gap between lab scenarios with abundant labeled data and real situations with scarce and unevenly distributed labeled data severely limits the practicality and scalability of FL. It motivates us to consider the following question: **How to make FL effective with scarce and skewly distributed labeled data as well as abundant unlabeled data?**

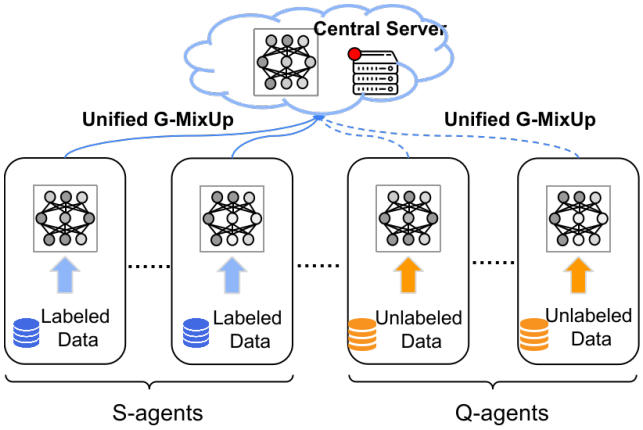

Figure 1: Overview of a distributed learning system with both labeled and unlabeled data sources, denoted with different colors. Our study can establish a shared global data space to boost SSL across all data sources, while keeping data privacy.

Semi-supervised learning (**SSL**) [Chapelle *et al.*, 2006] is an important machine learning topic that aims to leverage unlabeled data to enhance model capacity. The classical "manifold" assumption of centralized SSL [Zhou and Li, 2010] assumes that the data space is composed of multiple lower-dimensional manifolds, and data points lying on the same manifold should have the same label. A common practice [Sajjadi *et al.*, 2016] to ensure this assumption is to firstly augment each data instance with label-invariant transformation to multiple variants, then tune the models to classify them as the same label. Recent studies [Laine and Aila, 2017; Zhang *et al.*, 2018] further propose to perform soft data-label augmentations by mixing/blending both data features and la-

bels. **However, it is not clear how to perform effective SSL in the FL context, as data exchange is prohibited**.

To tackle this issue, we design an effective federated semi-supervised learning framework (FedSSL) to fully leverage both labeled and unlabeled data sources while keeping the data locality as FL. We propose to learn a global generative model to establish a unified data space across all data sources, enabling each agent to generate labeled data instances for local model training. We jointly optimize the objective of training the local model F to estimate accurate labels of the generated samples and the objective of training a generator G to provide realistic data imputations conditioning on inferred labels by F. To prevent training divergence with mass unlabeled data, we further regularize the model with self-reconstruction and realism maximization targets.

In addition, we aim to prevent privacy leakage for the labeled data sources. We design a hybrid training strategy of sequential and parallel training steps on labeled and unlabeled data sources respectively. This design integrates the differential privacy (DP) scheme in FedSSL smoothly to prohibit excessive access to the labeled data. We show that our strategy can protect privacy with strict theoretical guarantee, and it causes only minimal degradation in model performance.

In conclusion, our contributions include:

- We study a critical but under-explored task of effectively performing semi-supervised learning with distributed and mixed types of data sources.
- Our learning framework prevents the violation of data locality that existed in all previous studies of SSL in FL.
- We design a mixed-data generation strategy to utilize both labeled and unlabeled data sources by establishing a unified data space without direct data exchange.
- We firstly propose a private SSL framework in FL which ensures strict privacy protections for labeled data sources.
- We outperform baselines by 5%-15% on vision and NLP tasks and prevent divergence with extremely scarce data.

2 Related Work

We briefly review recent related work in categories: 1) semi-supervised learning (SSL), 2) federated learning (FL), and 3) settings involving both of them.

The SSL, e.g., [Chapelle *et al.*, 2006], is an important machine learning topic which aims to utilize unlabeled data to improve task learning. Classical SSL methods include pseudo-labeling [Lee and others, 2013] and entropy minimization [Grandvalet and Bengio, 2004]. Data augmentation approaches, such as MixUp [Zhang *et al.*, 2018], MixMatch [Berthelot *et al.*, 2019], and FixMatch [Sohn *et al.*, 2020], have also been developed and integrated into DL models. They interpolate pairs of data and label with random ratios to augment training data. However, existing approaches can only apply on centralized training paradigm, while our approach effectively applies on distributed data sources.

The FL has become a rapidly developing topic in the research community, e.g., [McMahan *et al.*, 2017; Zhao *et al.*, 2018; Li and others, 2019; Fan and Liu, 2020], as it provides a new way of learning models over a collection of distributed devices while keeping data locality. Recent studies focused on FL's robustness against non-IID data [Zhao *et al.*, 2018; Sattler *et al.*, 2019], few-shot data [Wu *et al.*, 2020; Fan and Huang, 2021] and differential privacy [Wei and others, 2020; Xin *et al.*, 2020]. However, these FL studies assume the agents in FL have plenty of labeled training data, while we focus on a practical scenario that labeled data is scarce.

Recently, several works made initial attempts to consider SSL in FL settings. Several surveys [Jin *et al.*, 2020a; Jin *et al.*, 2020b] discussed about applying existing SSL methods in FL without experimental proofs. Zhang *et al.* [2020] assumed that all labeled data is available at the server. Jeong *et al.* [2020] assumed that labeled data is available at *every* client. Itahara *et al.* [2020] assumed that *all* unlabeled data is shared across *all* agents. However, these studies violated the data locality property of FL. In contrast, we consider the scenario that labeled data or unlabeled data is kept at each client, and we prohibit sharing data across the agents.

3 Approach

We briefly review FL and SSL first, then formulate SSL objective in FL formally, and propose our FedSSL framework.

3.1 Review of Federated Learning

We consider a classical supervised FL system with K distributed agents. Each agent owns a local data source with which it trains a local learning model. A central server coordinates the agents by periodically collecting local models to fuse to a global model, then synchronizing back to all agents for next round of update.

Formally, let $\mathcal{X}_k$ be the data source of client k, n_k be the number of data samples in $\mathcal{X}_k$, $n = \sum_k n_k$ be the total number of samples across all data sources. We consider a C-class data space $\mathcal{D}$ with label space $\mathcal{Y} = \{0, ..., C\text{-}1\}$. Let F be the learning model with parameters w that maps data to label space $F : \mathcal{D} \to \mathcal{Y}$, e.g., a CNN for image classification.

The global FL target is to minimize the joint training objective $\mathcal{L}$ over all data sources $\min_w \mathcal{L}(w) = \sum_{k=1}^{K} \frac{1}{K} \ell_k(\mathcal{X}_k; w)$ in which local objective ℓ_k is the task-specific local training objective, e.g., cross-entropy loss for classification.

A widely used FL strategy called FedAvg [McMahan *et al.*, 2017] is to fuse the global model w with weighted average of local models such that $w = \sum_{k=1}^{K} \frac{n_k}{n} w_k$.

3.2 SSL Objective in FL

We consider a distributed SSL scenario where each learning agent owns one of two possible types of data sources, as shown in Figure 1. The first type of source owns a few labeled data instances (with no unlabeled data), and we call the agents with such source as support agents (**S-agents**). The other type of source owns all unlabeled data instances, and we call the agents with such source as query agents (**Q-agents**). The mission of our study is to *improve the collective capacity of both Q-agents and S-agents through SSL in FL paradigm*.

We consider a system of N_S S-agents and N_Q Q-agents. We denote the data collection for all S-agents as $\mathcal{S} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_{N_S}\}$, with each labeled source $\mathcal{X}_i = \{(\boldsymbol{x}_t, \boldsymbol{p}_t)\}_{t=1}^{|\mathcal{X}_i|}$. Here we use $\boldsymbol{p}_t \in \mathcal{R}^C$ as one-hot encoding

of the data label y_t. We denote the data collection for all Q-agents as $\mathcal{Q} = \{\mathcal{U}_1, \mathcal{U}_2, ..., \mathcal{U}_{N_Q}\}$, with each unlabeled data source as $\mathcal{U}_j = \{\boldsymbol{u}_t\}_{t=1}^{|\mathcal{U}_j|}$.

The global FL target is the joint of all local objectives on all S- and Q-agents as follows

$$\min_{w} \mathcal{L}(w) = \frac{1}{N_S}\sum_{i=1}^{N_S} \ell^s(\mathcal{X}_i; w) + \frac{1}{N_Q}\sum_{j=1}^{N_Q} \ell^q(\mathcal{U}_j; w) \ , \quad (1)$$

in which ℓ^s and ℓ^q are the task-specific loss functions on S-agent and Q-agent that we will explore later.

3.3 Local MixUp (L-MixUp) Operations

We introduce the local MixUp (L-MixUp) [Zhang *et al.*, 2018; Berthelot *et al.*, 2019] in classical SSL, for eliciting our design of global MixUp (G-MixUp) in the next section.

Pseudo Label. At Q-agent j with unlabeled data $\mathcal{U}_j$, we can guess the pseudo labels given a trained model w. Following [Berthelot *et al.*, 2019], we augment a data instance $\boldsymbol{u}_t \in \mathcal{U}_j$ to its K variants $\{\boldsymbol{u}_t^k\}_{k=1}^K$ with label-invariant operations, e.g., image rotation and cropping, and compute their mean probabilistic predictions $\bar{\boldsymbol{p}}_t = \frac{1}{K}\sum_{k=1}^K F(\boldsymbol{u}_t^k; w) \in \mathcal{R}^C$.

We can estimate the pseudo label by applying label sharpening [Berthelot *et al.*, 2019] to create low-entropy (sharp) label distribution $\hat{\boldsymbol{p}}_t$, with likelihood of each category c as

$$\hat{\boldsymbol{p}}_t^{(c)} = \text{Sharpen}(\bar{\boldsymbol{p}}_t, Z)_c = \frac{(\bar{\boldsymbol{p}}_t^{(c)})^Z}{\sum_{v=1}^C (\bar{\boldsymbol{p}}_t^{(v)})^Z} \ , \quad (2)$$

in which $Z > 1$ amplifies the dominating classes.

Local MixUp (L-MixUp) Operations. We describe the L-MixUp in details, which directly applies MixUp at each S- and Q-agent locally and individually to synthesize mixed data features and labels for local training.

Formally, for two data features $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with label (or pseudo label) distributions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, L-MixUp produces

$$\begin{aligned} \lambda &\sim \text{Beta}(\alpha, \alpha), \quad \alpha \in (0, 1) \ , \\ \boldsymbol{x}' &= \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2 \ , \\ \boldsymbol{p}' &= \lambda \boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2 \ , \end{aligned} \quad (3)$$

where $(\boldsymbol{x}', \boldsymbol{p}')$ is the synthesized data, and α is the mixup hyper-parameter which controls the strength of interpolation between two inputs. L-MixUp augments the local data set and regularizes the model towards producing linearly-behaving boundaries between classes thus makes the model predict more accurately for unlabeled data [Zhang *et al.*, 2018].

3.4 Global MixUp (G-MixUp) Operations

L-MixUp is limited as it can only operate locally, as data exchange among agents is prohibited in FL. This would lead to inferior performance in our challenging SSL setting of scarce and skewly distributed labeled data. To tackle this issue, we propose the global MixUp (G-MixUp) that operates across the agents to allow data imputation without data exchange.

G-MixUp utilizes a generative learning scheme to construct a global data space to generate and mix data of arbitrary class. We design a conditional generator G to synthesize a mixed data sample which conditions on both local data-label pair as well as an additional sampled data class from the global label space. We will discuss the details with S-agent and Q-agent individually.

G-MixUp at an S-agent.

Let the conditional generator G accept a labeled data with true label $(\boldsymbol{x}_1, \boldsymbol{p}_1)$, a randomly sampled class c_2 of one-hot form $\boldsymbol{p}_2$, a blending factor λ as in (3), and a noise vector $\boldsymbol{z} \in \mathcal{R}^{D_n}$. The goal of G is to produce a synthesized data $\hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{x}_1, \boldsymbol{p}_1), \boldsymbol{p}_2, \boldsymbol{z}, \lambda)$ with a predicted label $\hat{\boldsymbol{p}} = F(\hat{\boldsymbol{x}})$. Consistent with L-MixUp, the generated $\hat{\boldsymbol{x}}$ is supposed to be λ-likely as of $\boldsymbol{x}_1$'s class, and $(1-\lambda)$-likely as class c_2 in appearance, i.e., $\boldsymbol{p}' = \lambda \boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2$, which we can evaluate with the cross-entropy loss. Formally, the joint training objective for F and G is

$$\begin{aligned} &\min_{F,G} \ell^S = \text{CE}(\hat{\boldsymbol{p}}, \boldsymbol{p}') = -\sum_{c=1}^C p'(c) \cdot \log(\hat{p}(c)) \\ &s.t. \ \ \boldsymbol{p}_1, \boldsymbol{p}_2 \sim \mathcal{R}^C, \ \ \boldsymbol{z} \sim U(0,1)^{D_n}, \ \ \boldsymbol{x}_1 \sim \mathcal{X}_i \ , \\ &\qquad \hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{x}_1, \boldsymbol{p}_1), \boldsymbol{p}_2, \boldsymbol{z}, \lambda), \ \lambda \sim \text{Beta}(\alpha, \alpha) \ , \\ &\qquad \hat{\boldsymbol{p}} \leftarrow F(\hat{\boldsymbol{x}}) \ , \boldsymbol{p}' \leftarrow \lambda \boldsymbol{p}_1 + (1-\lambda)\boldsymbol{p}_2 \ . \end{aligned} \quad (4)$$

A better G produces more realistic imputations, and a better F produces more accurate estimation of the generated data labels. Hereby F and G improve each other to reach optimality. The global G can establish a global data space to facilitate effective global MixUp, thus improving SSL capacity.

G-MixUp at a Q-agent.

Next, we define G-MixUp at a Q-agent with unlabeled data. We firstly sample an unlabeled data $\boldsymbol{u}_1$, produce its pseudo label $\boldsymbol{q}_1$ with data augmentation and sharpening with (2). To perform mix-data generation, we then sample a new class c_2 with its one-hot form $\boldsymbol{p}_2$, draw a noise vector $\boldsymbol{z}$, and train G to produce a synthesized $\hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{u}_1, \boldsymbol{q}_1), \boldsymbol{p}_2, \boldsymbol{z}, \lambda)$ with a predicted label $\hat{\boldsymbol{p}} = F(\hat{\boldsymbol{x}})$.

Ideally, the synthesized data $\hat{\boldsymbol{x}}$ is expected to be λ-likely as $\boldsymbol{q}_1$ and $(1-\lambda)$-likely as $\boldsymbol{p}_2$ in appearance, i.e., $\boldsymbol{p}' = \lambda \boldsymbol{q}_1 + (1-\lambda)\boldsymbol{p}_2$, which we can evaluate with the cross-entropy loss. Formally, the joint training objective for F and G is

$$\begin{aligned} &\min_{F,G} \ell^Q = \text{CE}(\hat{\boldsymbol{p}}, \boldsymbol{p}') = -\sum_{c=1}^C p'(c) \cdot \log(\hat{p}(c)) \\ &s.t. \ \ \boldsymbol{p}_2 \sim \mathcal{R}^C, \ \ \boldsymbol{z} \sim U(0,1)^{D_n}, \ \ \boldsymbol{u}_1 \sim \mathcal{U}_j \ , \\ &\qquad \boldsymbol{q}_1 = \text{Sharpen}(F(\text{Aug}(\boldsymbol{u}_1)), Z) \ , \\ &\qquad \hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{u}_1, \boldsymbol{q}_1), \boldsymbol{p}_2, \boldsymbol{z}, \lambda), \ \lambda \sim \text{Beta}(\alpha, \alpha) \ , \\ &\qquad \hat{\boldsymbol{p}} \leftarrow F(\hat{\boldsymbol{x}}) \ , \boldsymbol{p}' \leftarrow \lambda \boldsymbol{q}_1 + (1-\lambda)\boldsymbol{p}_2 \ . \end{aligned} \quad (5)$$

We further regularize model training by ensuring the realism of generated samples and reconstructed real samples.

Realistic loss. Equations (4) and (5) imply that if we set the blending factor $\lambda = 0$, G will generate a data sample 100% of class c_2. We can utilize a conditional discriminator D to encourage realism of the generated image with given class.

We extend to a general case of $\lambda \in [0, 1]$, to encourage the realism of G output weighed by a factor $g(\lambda)$ depending on

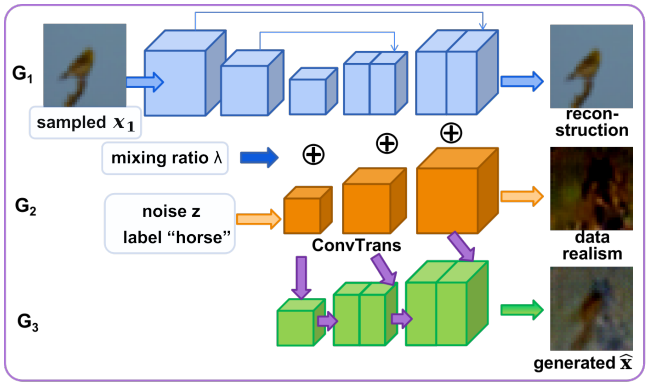

Figure 2: Model design for model G.

Algorithm 1: FedSSL algorithm.

Input: A set of N_S support agents, and N_Q query agents.
Output: A global model F for SSL task.

1 **Server executes:** Initialize global F, G, D, let $t \leftarrow 1$
2 **while** $t \leq$ *maximum rounds* T **do**
3 **for** each client i *in support set* $\mathcal{S}$ **in paral**
4 $[D, G]_i^t \leftarrow$ G-Reg$([D, G]^t, \mathcal{X}_i)$
5 $[F, G]_i^t \leftarrow$ G-MixUp$([F, G]^t, \mathcal{X}_i)$
6 **for** each client j *in query set* $\mathcal{Q}$ **in parallel do**
7 $[D, G]_j^t \leftarrow$ G-Reg$([D, G]^t, \mathcal{U}_j)$
8 $[F, G]_j^t \leftarrow$ G-MixUp$([F, G]^t, \mathcal{U}_j)$
9 $[F, D, G]^{t+1} \leftarrow FedAvg(\{[F, D, G]_{k\in\mathcal{S}\cup\mathcal{Q}}^t\})$
10 The server sends $[F, D, G]^{t+1}$ back to clients
11 $t \leftarrow t + 1$

1, train G and D based on synthetic data, 优化生成模型G和真实判别模型D
2, Global mixup, 优化生成模型G和增强样本判别模型F

blending factor λ. We design the *dynamic realistic loss* as

$$\begin{aligned} &\min_G \max_D \ell^{real} = \log D(\boldsymbol{x}_1, \boldsymbol{p}_1) + g(\lambda)\log(1 - D(\hat{\boldsymbol{x}}, \boldsymbol{p}_2)) \\ &s.t.\ \ \boldsymbol{p}_1, \boldsymbol{p}_2 \sim \mathcal{R}^C,\ \ \boldsymbol{z} \sim U(0,1)^{D_n},\ \ \boldsymbol{x}_1 \sim \mathcal{X}^{c_1}\ , \\ &\quad \hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{x}_1, \boldsymbol{p}_1), \boldsymbol{p}_2, \boldsymbol{z}, \lambda),\ \lambda \in [0, 1]\ , \\ &\quad g(\lambda) \leftarrow e^{\max\{\lambda, 1-\lambda\}-1}\ , \end{aligned} \tag{6}$$

in which $\max\{\lambda, 1-\lambda\} - 1 \in [-0.5, 0]$ and $g(\lambda) \in [0.61, 1]$. When blending from one source of either sampled image or noise ($\lambda = 0, 1$), we encourage more visually realism with large $g(\lambda) = 1$; when blending evenly from two sources ($\lambda = 0.5$), we tolerate the unrealism with a smaller $g(\lambda) = 0.61$. We adopt the training process of GAN [Goodfellow *et al.*, 2014] to update D and G alternatively.

Reconstruction loss. When setting the blending factor $\lambda = 1$, Equations (4) and (5) imply that G should re-produce the sampled $\boldsymbol{x}_1$ (or $\boldsymbol{u}_1$). Thus we design an encoder-decoder structure of G to reconstruct the input, and we regularize its training with reconstruction loss term such that

$$\begin{aligned} &\min_G \ell^{rec} = ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_2^2 \\ &s.t.\ \ \hat{\boldsymbol{x}} \leftarrow G((\boldsymbol{x}_1, \boldsymbol{p}_1), c_2, \boldsymbol{z}, \lambda = 1)\ . \end{aligned} \tag{7}$$

FedSSL Algorithm Details. We show our proposed FedSSL with G-MixUp in Algorithm 1 and summarize as follows. At local training stage of FL, we adopt a two-step optimization procedure to 1) regularize model by training D and G alternatively to minimize $\ell^{reg} = \ell^{real} + \ell^{rec}$, which we denote as *G-Reg* (line 4,7); 2) train F and G with *G-MixUp* to produce mixed data samples and improve accuracy (line 5,8). By federating the models (line 9), we *unify G-MixUp operations on S-and Q-agents* to leverage both true labels and pseudo labels to enhance SSL, and *establish a unified global data space* over all agents to facilitate data augmentation over entire data space with arbitrary classes, thus boost SSL.

Algorithm 2: Pseudo code for FedSSL-DP.

1 Let $\epsilon_{tot} \leftarrow 0,\ \delta \leftarrow 10^{-5}$.
2 **for** each client i *in support set* $\mathcal{S}$ **in sequence do**
3 **if** $\epsilon_{tot} > \epsilon$ **then**
4 *Out of privacy budget, break;*
5 **else**
6 *Perform DP update on* G;
7 $\epsilon_{\text{tot}} \leftarrow \epsilon_{\text{tot}} + \alpha_{\mathcal{M}}(\sigma, \delta)$;
8 *Perform normal update on* D, F;
9 **for** each client j *in query set* $\mathcal{Q}$ **in parallel do**
10 *Perform normal update on* D, F, G;

4 Privacy

In the realistic setting as we consider, the small labeled data could be overly accessed by G during G-MixUp, which poses risks of information leakage (as the global G could reconstruct them at arbitrary agents). In the spirit of FL for privacy protection, it's critical for us to ensure privacy for G-MixUp operations. In this section, we introduce (ϵ, δ)-DP to our framework and provide a practice algorithm which integrates noise injection mechanism into FedSSL to provide strict privacy guarantee for labeled data sources.

Definition 1. *(ϵ, δ)-DP [Dwork and Roth, 2014]. A randomized function $M : D \to \mathbb{R}$, with domain D and range R, satisfies (ϵ, δ)-DP if for any two adjacent databases $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq R$*

$$P[M(d) \in S] \leq e^{\epsilon} P[M(d') \in S] + \delta\ .$$

A standard way to guarantee (ϵ, δ)-DP is by integrating a Gaussian Mechanism (GM) [Dwork and Roth, 2014] in the model learning process, by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$. Stochastic GM (SGM) [Abadi *et al.*, 2016] is developed to work with SGD for training deep-learning models. SGM applies GM on sampled batches with a sampling ratio γ, and T update steps of SGM will imply a $(O(\sqrt{T}\gamma\epsilon, \delta)$-DP by using Moment Accountant (MA)[Abadi *et al.*, 2016] to track the overall privacy budget.

We propose a practical algorithm called FedSSL-DP, that seamlessly integrates SGM and MA into FedSSL to ensure (ϵ, δ)-DP , with pseudo code shown below.

Overall, we design a hybrid training strategy to train on S-agents and Q-agents efficiently. Specifically, we train on S-agents with DP **sequentially** first (line 2), i.e., train on S-agent i, then pass the updated model to S-agent $i + 1$. After looping over all S-agents, we perform the standard FL **in parallel** with Q-agents (line 9).

We first fix the tolerance term δ at a small value, e.g., 10^{-5}; then choose a target privacy budget upper bound ϵ, e.g., 4, 8, or 16, and initialize accumulator ϵ_{tot} (line 1). Then we perform DP update on G with SGM, then calculate ϵ as privacy loss with MA $\alpha_{\mathcal{M}}(\sigma, \delta)$ and accumulate to ϵ_{tot} (line 6-8). Once ϵ_{tot} reaches the budget ϵ, we stop accessing to labeled sources (line 3-4).

Lemma 1. *Algorithm 2 guarantees (ϵ, δ)-DP for labeled data sources.*

Proof. Algorithm 2 guarantees that the cumulative privacy loss for updating G during sequential S-agent updates $\epsilon_{\text{tot}} \leq \epsilon$. Also, the parallel Q-agent update does not access the labeled data on S-agents. Therefore, G is (ϵ, δ)-DP with respect to labeled data sources. □

5 Experiments and Discussions

We describe the datasets, parameter choices and models we experiment on. Then we analyse the performance with visual and textual tasks with ablation studies and visualizations.

5.1 Datasets and Splits

Following recent works [Berthelot *et al.*, 2019; Zhu *et al.*, 2020; Chen and et al., 2018] of evaluating FL and SSL, we describe three widely used benchmark datasets.

CIFAR-10 [Krizhevsky, 2009] is a common image recognition dataset with 50000 data instances of 10 categories (such as birds, cars, and horses). We try 3 settings with an increasing difficulty, by **holding out 5000 (10%), 2500 (5%), and 500 (1%) as labeled instances**, respectively, and keeping the rest as unlabeled instances.

MNIST [LeCun *et al.*, 1998] is a digit recognition dataset with 60000 data instances for 10 digital classes. We try 3 settings with an increasing difficulty, by **holding out about 300 (0.5%), 150 (0.25%), and 100 (0.17%) of total data as labeled instances** respectively, and using the rest as unlabeled.

Sent140 [Caldas *et al.*, 2018] is an FL benchmark for sentiment analysis as a 2-way classification task (positive and negative). We sample 60,000 sentences and try 3 settings with increasing difficulty by **holding out 3000 (5%), 600 (1%), and 300 (0.5%) as labeled sentences**, with the rest unlabeled. We tokenize each sentence to a max of 40 words.

5.2 FL Device Numbers

We try different numbers of S-agents (labeled sources) and Q-agents (unlabeled sources), denoted as N_S and N_Q respectively. We examine two different settings, i.e., $(N_S = 2, N_Q = 6)$, and $(N_S = 3, N_Q = 9)$, to represent the scenario of fewer S-agents and more Q-agents. We indeed try extremely challenging cases, e.g., 100 (0.17%) MNIST samples distributed to 3 S-agents so that each has about 34 labeled samples, for checking whether FedSSL is robust to extreme data-scarce scenarios.

5.3 Details of Our Methods and Baselines

We compare our proposed methods (FedSSL and FedSSL-DP) with the most related baselines as follows:

FedSSL is our method which learns a global data space to generate mixed samples of arbitrary classes to better augment local training data, as in Alg. 1. **FedSSL-DP** additionally integrates Gaussian Mechanism (GM) into FedSSL to ensure DP of labeled data sources, as in Alg. 2.

Baselines. Supervise performs FL only at S-agents with labeled data. Q-agents (with unlabeled data) are not used. **Pseudo** performs additional SSL with pseudo labels (Sec.3.3) at Q-agents upon *Supervise* approach. **MixMatch** [Berthelot *et al.*, 2019] is the commonly adopted centralized SSL approach, described as the L-MixUp in Section 3.3. We applied MixMatch on all agents to utilize both labeled and unlabeled data as a fair baseline. FixMatch [Sohn *et al.*, 2020] is the augmentation of MixMatch. Its FL version – **FedMatch** [Jeong *et al.*, 2020] – performs both L-MixUp and divergence minimization across all agents, which forms the strongest baseline.

5.4 Experimental Results

Results on CIFAR-10.

Method \ Setting	$N_S = 2, N_Q = 6$			$N_S = 3, N_Q = 9$		
	10%	5%	1%	10%	5%	1%
Supervise	0.754	0.654	0.344	0.759	0.633	0.302
Pseudo	0.792	0.727	0.549	0.782	0.743	0.537
MixMatch	0.822	0.769	0.558	0.811	0.757	0.548
FedMatch	0.803	0.747	0.568	0.785	0.753	0.547
FedSSL-DP(ours)	0.848	**0.793**	0.634	0.839	0.777	0.628
FedSSL (ours)	**0.855**	**0.801**	**0.661**	**0.854**	**0.787**	**0.653**

Table 1: CIFAR-10 results of 10-way classification accuracy.

We evaluate on CIFAR-10 with randomly sampled 10%, 5%, and 1% of the total training data as labeled data, which accounts for roughly 5000, 2500, and 500 total training samples, respectively. We uniformly distribute the labeled data to S-agents, and unlabeled data to Q-agents. Table 1 summarizes the results for 8 and 12 devices:

FedSSL consistently performs best in all settings of labeled data portion and client number. For 10% labeled data setting, FedSSL achieves the best accuracy of 0.855 and 0.854, outperforming the best baseline (MixMatch or FedMatch) by 4.2-5.3%, relatively. For 1% labeled data, the relative performance gain reaches 14.1%-19.2%. This shows the effectiveness of FedSSL especially in extreme data-scarce conditions.

Baselines are ineffective. Compared with *Supervise*, *Pseudo* labels could help improve performance by 3%, 6%, and 15% in absolute value for 10%, 5%, and 1% label data respectively. However, L-MixUp (FedMatch and MixMatch) could only further increase performance by less than 3% in absolute value, indicating the ineffectiveness of conventional SSL techniques, due to the skewly distributed data labels.

FedSSL-DP achieves comparable performance with FedSSL. For 10% and 5% labeled data cases, the performance gap is less than 2%. The biggest differences happen for 8 and 12 devices with 1% labeled data, which are about 2.5-2.8% in absolute value. DP affects the model accuracy more with less labels and larger device numbers.

FedSSL-DP outperforms the baselines significantly, indicating that our approach is effective in distributed learning while preserving the privacy of the labeled sources.

Results on MNIST.

Method \ Setting	$N_S=2, N_Q=6$			$N_S=3, N_Q=9$		
	0.5%	0.25%	0.17%	0.5%	0.25%	0.17%
Supervise	0.951	0.889	-	0.951	-	-
Pseudo	0.966	0.939	-	0.969	-	-
MixMatch	0.979	0.957	-	0.977	-	-
FedMatch	0.985	0.962	-	0.981	-	-
FedSSL-DP(ours)	0.981	0.965	0.950	**0.986**	**0.972**	0.949
FedSSL(ours)	**0.988**	**0.970**	**0.976**	**0.987**	**0.975**	**0.969**

Table 2: MNIST results of 10-way classification accuracy.

We evaluate our methods on MNIST with 0.5%, 0.25% and 0.17% of the total data as labeled data, which accounts for just 300, 150 and 100 total training samples, respectively. We show the results in Table 2:

The baselines suffer from model divergence. In the extreme cases of training with 0.25% and 0.17% labeled data on 8 (N_S=2, N_Q=6) and 12 (N_S=3, N_Q=9) devices, all baselines failed to converge and predict randomly (denoted as '-'). Due to lack of a unified data space, each local model overfits to local data so that the federated global model collapses.

FedSSL and FedSSL-DP can prevent divergence. In contrast, our proposed FedSSL achieves reasonable accuracy of around 0.97. Thanks to the global data space, FedSSL augments local data and prevents overfitting to scarce labels (S-agents) and incorrect pseudo labels (Q-agents).

FedSSL-DP achieves similar accuracy with FedSSL, with performance gap generally under 1% in absolute value. Even for extreme data-scarce (0.17%) case, the gap is below 3%, indicating the usability of FedSSL-DP.

Results on Text Classification.

In Table 3, we show results on Sent140 dataset with 5%, 1% and 0.5% of the total tweets with sentiment labels, which accounts for roughly 3000, 600, and 300 sentences respectively. We implement blending operation of two sentences by a simple weighted sum of two words' BERT embeddings from two sentences at same positions. Table 3 shows the binary classification results. *FedSSL consistently outperforms the baselines*, leading the next best FedMatch by 1.8-5% relatively, while leading the weakest *Supervise* by 7-12% relatively.

Setting	$N_S=1, N_Q=3$			$N_S=2, N_Q=6$		
	5%	1%	0.5%	5%	1%	0.5%
Supervise	0.700	0.648	0.628	0.690	0.639	0.622
Pseudo	0.733	0.672	0.660	0.721	0.671	0.638
FedMatch	0.741	0.691	0.670	0.724	0.683	0.661
FedSSL(ours)	**0.754**	**0.722**	**0.699**	**0.749**	**0.703**	**0.689**

Table 3: Sent140 results of 2-way classification accuracy.

5.5 Ablation Studies

Non-IID data partition. We consider a more challenging setting with non-IID partition of data classes. We adopt a round robin strategy of distributing non-overlapping MNIST digital classes to each agent. As an example of ($N_S = 3, N_Q = 9$), we firstly distribute all *labeled* instances of digital classes $[0, 3, 6, 9]$, $[1, 4, 7]$, $[2, 5, 8]$ to 3 S-agents respectively, then we distribute all *unlabeled* instances of $[0, 9], [1], [2], ..., [8]$ to 9 Q-agents respectively. This creates non-overlapping partitions of digital classes within S-agent group and Q-agent group thus makes local training header.

Setting	$N_S=1/2/3, N_Q=3/6/9$	
	0.25%	0.17%
Supervise	0.854 / 0.868 / -	0.798 / - / -
Pseudo	0.907 / 0.917 / -	0.849 / - / -
MixMatch	0.927 / 0.925 / -	0.864 / - / -
FedMatch	0.932 / 0.927 / -	0.869 / - / -
FedSSL	**0.964 / 0.949 / 0.665**	**0.952 / 0.801 / 0.551**

Table 4: MNIST results in Non-IID settings.

We examine the performance of baselines and our models under this difficulty situation and observe in Table 4 that:

The non-IID settings bring about performance drop and divergence especially for scarce labeled data 0.25% and 0.17% with a large device number 8 and 12.

Only FedSSL could perform reasonably well in extreme settings (e.g., 0.17% labeled data), while all other baselines diverge due to collapsed local training on partial data classes.

General setting of partially labeled data. FedSSL can readily extend to a general setting in which each client has both labeled and unlabeled data, as our proposed G-MixUp can flexibly sample data pairs with true labels as Eq.(4) and/or with pseudo labels as Eq.(5). We allocate 10% of labeled images and the rest unlabeled images of CIFAR-10 uniformly to 4 clients. FedSSL outperforms the baseline FedMatch by 5.9% (85.1% v.s. 79.2%), as our G-MixUp can perform both local and global data imputation to better train the unified global model, while FedMatch can only perform local mixup.

# labeled	FedSSL(Tab.4)	no l^{rec}	no l^{real}	w/o both
0.25%	96.4%	-0.83%	-0.93%	-1.8%
0.17%	95.2%	-1.3%	-2.8%	**-29.3%**

Table 5: Ablation study of reconstruction and realistic loss.

Effects of l^{real} (Eq.6) and l^{rec} (Eq.7). We evaluate FedSSL on non-iid MNIST with 3 ablation settings: no realistic loss l^{rec}, no reconstruction loss l^{real} and without both. We find that both l^{rec} and l^{real} are critical with better regularization especially for the extreme data-scarce scenario (0.17%), e.g., w/o both would yield 29.3% drop of accuracy.

6 Conclusion

We proposed a unified framework that makes FL effective in challenging SSL scenarios. We designed a generative learning strategy to establish a global data space across the agents while preserving data privacy with theoretical guarantee. Our approach outperforms the baselines significantly and works robustly in extreme data-scarce and non-IID cases.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC 62106156), Shenzhen Science and Technology Program (Project JCYJ20210324120011032), Guangdong Basic and Applied Basic Research Foundation (Project 2021B1515120008), and the Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *SIGSAC*, 2016.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019.

[Caldas *et al.*, 2018] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecnỳ, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[Chapelle *et al.*, 2006] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. 2006.

[Chen and et al., 2018] Fei Chen and et al. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[Fan and Huang, 2021] Chenyou Fan and Jianwei Huang. Federated few-shot learning with adversarial learning. *arXiv preprint arXiv:2104.00365*, 2021.

[Fan and Liu, 2020] Chenyou Fan and Ping Liu. Federated generative adversarial learning. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 2020.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[Grandvalet and Bengio, 2004] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004.

[Itahara *et al.*, 2020] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *arXiv preprint arXiv:2008.06180*, 2020.

[Jeong *et al.*, 2020] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency. *arXiv preprint arXiv:2006.12097*, 2020.

[Jin *et al.*, 2020a] Yilun Jin, Xiguang Wei, Yang Liu, and Qiang Yang. A survey towards federated semi-supervised learning. *arXiv preprint arXiv:2002.11545*, 2020.

[Jin *et al.*, 2020b] Yilun Jin, Xiguang Wei, Yang Liu, and Qiang Yang. Towards utilizing unlabeled data in federated learning: A survey and prospective. *arXiv preprint arXiv:2002.11545*, 2020.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[Laine and Aila, 2017] Samuli Matias Laine and Timo Oskari Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

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

[Li and others, 2019] Tian Li et al. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[Sajjadi *et al.*, 2016] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NIPS*, 2016.

[Sattler *et al.*, 2019] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE TNNLS*, 2019.

[Sohn *et al.*, 2020] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A. Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[Wei and others, 2020] Kang Wei et al. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020.

[Wu *et al.*, 2020] Qiong Wu, Kaiwen He, and Xu Chen. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *IEEE Computer Graphics and Applications*, 2020.

[Xin *et al.*, 2020] Bangzhou Xin, Wei Yang, Yangyang Geng, Sheng Chen, Shaowei Wang, and Liusheng Huang. Private fl-gan: Differential privacy synthetic data generation based on federated learning. In *ICASSP*, 2020.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[Zhang *et al.*, 2020] Zhengming Zhang, Yaoqing Yang, Zhewei Yao, Yujun Yan, Joseph E Gonzalez, and Michael W Mahoney. Improving semi-supervised federated learning by reducing the gradient diversity of models. *arXiv preprint arXiv:2008.11364*, 2020.

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

[Zhou and Li, 2010] Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 2010.

[Zhu *et al.*, 2020] Jianchao Zhu, Liangliang Shi, Junchi Yan, and Hongyuan Zha. Automix: Mixup networks for sample interpolation via cooperative barycenter learning. *ECCV*, 2020.