# Federated Semi-Supervised Learning with Inter-Client Consistency

**Wonyong Jeong** [1] **Jaehong Yoon** [1] **Eunho Yang** [1 2] **Sung Ju Hwang** [1 2]

## Abstract

While existing federated learning approaches mostly require that clients have fully-labeled data to train on, in realistic settings, data obtained at the client side often comes without any accompanying labels. Such deficiency of labels may result from either high labeling cost, or difficulty of annotation due to requirement of expert knowledge. Thus the private data at each client may be only partly labeled, or completely unlabeled with labeled data being available only at the server, which leads us to a new problem of *Federated Semi-Supervised Learning (FSSL)*. In this work, we study this new problem of semi-supervised learning under federated learning framework, and propose a novel method to tackle it, which we refer to as *Federated Matching (FedMatch)*. FedMatch improves upon naive federated semi-supervised learning approaches with a new inter-client consistency loss and decomposition of the parameters into parameters for labeled and unlabeled data. Through extensive experimental validation of our method in two different scenarios, we show that our method outperforms both local semi-supervised learning and baselines which naively combine federated learning with semi-supervised learning.

## 1. Introduction

*Federated Learning (FL)* (McMahan et al., 2017; Zhao et al., 2018; Li et al., 2018; Chen et al., 2019a;b), in which multiple clients collaboratively learn a global model via coordinated communication, has been an active topic of research over the past few years. The most distinctive difference of federated learning from distributed learning is that the data is only *privately accessible* at each local client, without inter-client data sharing. Such decentralized learning brings us numerous advantages in addressing real-world issues such as data privacy, security, and access rights. For example, for on-device learning of mobile devices, the service provider may not directly access local data since they may contain privacy-sensitive information. In healthcare domains, the hospitals may want to improve their clinical diagnosis systems without sharing the patient records.

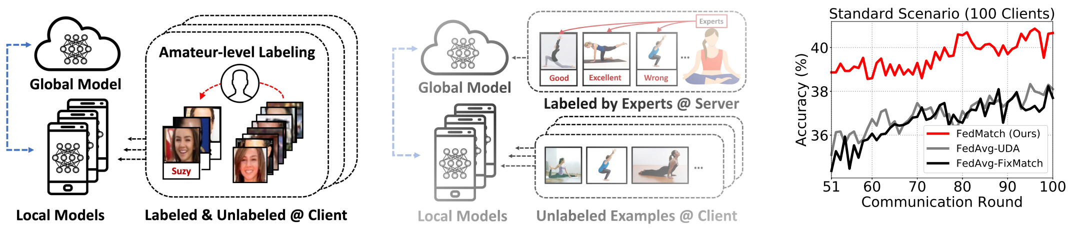
Existing federated learning approaches handle these problems by aggregating the locally learned model parameters. A common limitation is that they only consider supervised learning settings, where the local private data is fully labeled. Yet, the assumption that all of the data examples may include sophisticate annotations is not realistic. Suppose that we perform on-device federated learning, the users may not want to spend their time and efforts in annotating the data, and the participation rate across the users may largely differ. Even in the case of enthusiastic users may not be able to fully label all the data in the device, which will leave the majority of the data as unlabeled (See Figure 1 (a)). Moreover, in some scenarios, the users may not have sufficient expertise to correctly label the data. Suppose that we have a workout app that automatically evaluates and corrects one's body posture. In this case, the end users may not be able to evaluate his/her own body posture at all. Thus, in many realistic scenarios for federated learning, local data will be mostly *unlabeled*. This leads us to a new problem of *Federated Semi-Supervised Learning (FSSL)*.

A naive solution to this federated semi-supervised learning is to simply perform semi-supervised learning (SSL) using any off-the-shelf methods (e.g. FixMatch (Sohn et al., 2020), UDA (Xie et al., 2019)) with federated learning algorithms to aggregate the learned weights. Yet, this does not fully exploit the knowledge of the multiple models trained on heterogeneous data.

To address this problem, we present a novel framework, *Federated Matching (FedMatch)*, which enforces the consistency between the predictions made across multiple models. Further, we decompose the model parameters into two, one for supervised and another for unsupervised learning, where the former is dense and the latter is sparse. This

[1]Korea Advanced Institution of Science and Technology, South Korea [2]AITRICS, South Korea. Correspondence to: Wonyong Jeong <wyjeong@kaist.ac.kr>, Sung Ju Hwang <sjhwang82@kaist.ac.kr>.

*International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2020*, Vienna, Austria, PMLR 108, 2020. 




(a) Standard Scenario (b) Disjoint Scenario (c) Our Methods

*Figure 1.* **Concept Illustrations for Federated Semi-Supervised Learning Scenarios and Our Methods for FSSL** (a) describes *Standard Scenario*, where both labeled and unlabeled instances are available at client. (b) represents *Disjoint Scenario*, where labeled instances are available only at server while unlabeled examples are given to local clients. (c) shows performance comparison between naive federated SSL models and our novel proposed scheme, FedMatch, with 100 clients on Batch IID Dataset (CIFAR-10).

sparse additive parameter decomposition ensures that training on labeled and unlabeled data are effectively separable, thus minimizing interference between the two tasks. Also, by utilizing sparse weights to for unlabeled tasks, we could significantly reduce the cost in communicating model parameters between clients for consistency regularization. We validate FedMatch on both scenarios of FSSL (Figure 1(a) and 1(b)) and show that our models significantly outperform baselines, including a naive combination of federated learning with semi-supervised learning (See Figure 1(c)), on the training data which are distributed non-i.i.d. and streams into the clients as in most realistic scenarios. The main contributions of this work are as follows:

- We introduce a **novel problem** of **Federated Semi-Supervised Learning (FSSL)** to tackle realistic federated learning scenarios where the local data is partly labeled or unlabeled.
- We propose a **novel framework** for FSSL, **Federated Matching (FedMatch)**, which learns for unlabeled data by maximizing the agreement between models trained on multiple clients, and performs sparse additive decomposition of model parameters to reduce both interference between supervised and unsupervised tasks, and communication cost.
- We experimentally validate that our **FedMatch significantly outperforms** both single-client SSL and the naive combination of SSL with federated learning algorithms under two realistic scenarios for FSSL.

## 2. Federated Semi-Supervised Learning

We introduce a realistic federated learning scenario, *Federated Semi-Supervised Learning* (FSSL). We first formally define the conventional semi-supervised learning and federated learning. Then, we define a federated semi-supervised learning and elaborate on two possible scenarios for the problem.

### 2.1. Preliminaries

**Semi-Supervised Learning** *Semi-Supervised Learning (SSL)* refers to the problem of learning with partially labeled data, where the ratio of unlabeled data is usually much larger than that of the labeled data (e.g. 1:9). Let $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ be a given dataset, where $\mathbf{x}_i$ is an arbitrary training instance with a corresponding one-hot label $\mathbf{y}_i \in \{1, \ldots, C\}$ for the $C$-way multi-class classification problem and $N$ is the number of instances. For SSL, $\mathcal{D}$ is further split into labeled and unlabeled data. Let $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^S$ be a set of $S$ labeled data instances and $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^U$ be a set of $U$ unlabeled samples without corresponding label. Here, in general, $|\mathcal{S}| \ll |\mathcal{U}|$. With these two datasets, $\mathcal{S}$ and $\mathcal{U}$, we now perform semi-supervised learning. Let $p_\theta(\mathbf{y}|\mathbf{x})$ be a neural network that is parameterized by weights $\boldsymbol{\theta}$ and predicts softmax outputs $\hat{\mathbf{y}}$ with given input $\mathbf{x}$. Our objective is to minimize loss function $\ell_{final}(\boldsymbol{\theta}) = \ell_s(\boldsymbol{\theta}) + \ell_u(\boldsymbol{\theta})$, where $\ell_s(\boldsymbol{\theta})$ is loss term for supervised learning on $\mathcal{S}$ and $\ell_u(\boldsymbol{\theta})$ is loss term for unsupervised learning on $\mathcal{U}$.

**Federated Learning** Federated Learning aims to collaboratively learn a global model via coordinated communication with multiple clients. Let $G$ be a global model and $\mathcal{L} = \{l_k\}_{k=1}^K$ be a set of local models for $K$ clients. $\mathcal{D}$ is composed of $K$ sub-datasets $\mathcal{D}^{l_k} = \{\mathbf{x}_i^{l_k}, \mathbf{y}_i^{l_k}\}_{i=1}^{N^{l_k}}$ privately spread to each client or local model $l_k$. At each communication round $r$ of training, global model $G$ randomly selects the local models that are available for training $\mathcal{L}^r \subset \mathcal{L}$. Then, $G$ initializes $\mathcal{L}^r$ with global weights $\boldsymbol{\theta}^G$, and the active local models $l_a \in \mathcal{L}^r$ perform supervised learning to minimize loss $\ell_s(\boldsymbol{\theta}^{l_a})$ on the corresponding sub-dataset $\mathcal{D}^{l_a}$. After that, $G$ aggregates the learned weights $\boldsymbol{\theta}^G \leftarrow \frac{1}{|\mathcal{L}^r|}\sum_a \boldsymbol{\theta}^{l_a}$ and broadcasts newly aggregated weights to local models that would be available at the next round $r+1$, and repeat the learning procedure until the final round $R$.

### 2.2. Federated Semi-Supervised Learning

Now we further describe the semi-supervised learning problems under federated learning framework, which we refer to as *Federated Semi-Supervised Learning*, in which the data obtained at the clients may or may not come with accompanying labels. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, $\mathcal{D}$

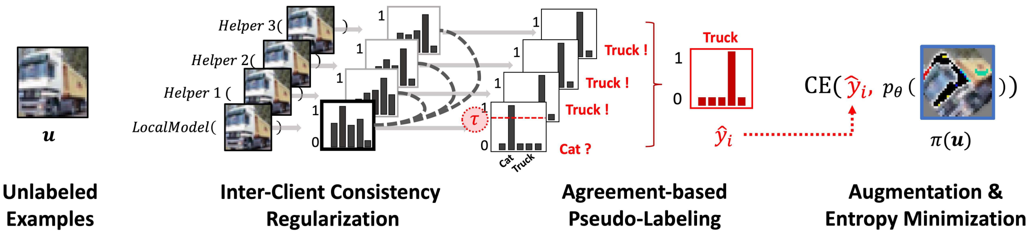



*Figure 2.* **Illustration of FedMatch Algorithm** Given unlabeled instance $\mathbf{u}$, we perform inter-client consistency regularization, which enforces consistency for the same input across different *models*. Then, we decide pseudo-label $\hat{\mathbf{y}}$ on certain class, of which probability is higher than threshold $\tau$, and also agreed by helper agents. At last, we perform entropy minimization with $\hat{\mathbf{y}}$ and perturbed image $\pi(\mathbf{u})$.

is split into a labeled set $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{S}$ and a unlabeled set $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^{U}$ as in the standard semi-supervised learning. Under the Federated Learning framework, we have a global model $G$ and a set of local models $\mathcal{L}$ where the unlabeled dataset $\mathcal{U}$ is privately spread over $K$ clients hence $\mathcal{U}^{l_k} = \{\mathbf{u}_i^{l_k}\}_{i=1}^{U^{l_k}}$. For a labeled set $\mathcal{S}$ on the other hand, we consider two different scenarios depending on the availability of labeled data at clients, namely the *standard scenario* (labeled data available at each client) and the *disjoint scenario* (labeled data only available at server).

**Standard Scenario** The standard scenario posits that the end-users intermittently annotate a small portion of their local data (i.e., $5\%$ of the entire data), while the rest of data instances remains unlabeled. This is a common scenario for user-generated personal data, where the end-users can easily annotate the data but may not have time or motivation to label all the data. We further assume that there is no server-side training, in which case the clients train on both labeled and unlabeled data, while the server only aggregates the updates from the clients and redistributes the aggregated parameters back to the clients, as illustrated in Figure 1 (a). In this scenario, labeled data $\mathcal{S}$ can be rewritten using individual sub-dataset $\mathcal{S}^{l_k} = \{\mathbf{x}_i^{l_k}, \mathbf{y}_i^{l_k}\}_{i=1}^{S^{l_k}}$, yielding $K$ sub-datasets for $K$ local models $l_{1:K}$. The overall learning procedure of the global model is the same as that of conventional federated learning (global model $G$ aggregates updates from the selected subset of clients and broadcasts them), except that active local models $l_{1:A}$ perform semi-supervised learning by minimizing loss $\ell_{final}(\boldsymbol{\theta}^{l_a}) = \ell_s(\boldsymbol{\theta}^{l_a}) + \ell_u(\boldsymbol{\theta}^{l_a})$ respectively on $\mathcal{S}^{l_a}$ and $\mathcal{U}^{l_a}$ rather than performing supervised learning. We refer to this scenario as the *standard scenario*, because local model $l_k$ perform standard semi-supervised learning.

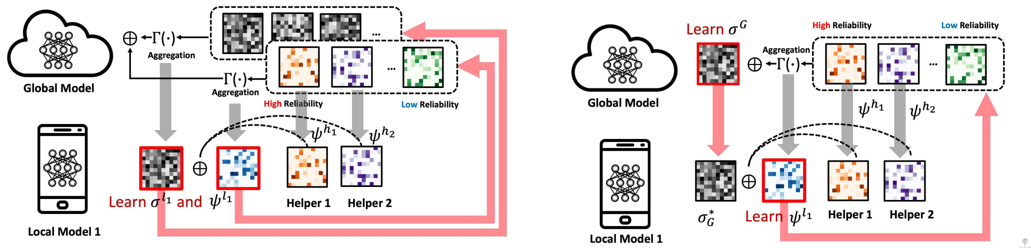
**Disjoint Scenario** This scenario assumes that the supervised labels are only available at the server, while local clients work with unlabeled data as described in Figure 1 (b). This is a common case for real-world applications where labeling requires expert knowledge (e.g. annotating medical images, evaluating body postures for exercises), but the data cannot be shared due to privacy concerns. In this scenario, $\mathcal{S}^G$ is identical to $\mathcal{S}$ and is located at server. The overall learning procedure is the same as that of federated learning, except the global model $G$ performs supervised learning on $\mathcal{S}^G$ by minimizing the loss $\ell_s(\boldsymbol{\theta}^G)$ before broadcasting $\boldsymbol{\theta}^G$ to local clients. Then, the active local clients $l_{1:A}$ at communication round $r$ perform unsupervised learning which solely minimizes $\ell_u(\boldsymbol{\theta}^{l_a})$ on the unlabeled data $\mathcal{U}^{l_a}$. We refer to this scenario as the *disjoint scenario* as the learning procedures with labeled and unlabeled data are disjointly done at the clients and the server, respectively.

### 2.3. Federated Matching

**Inter-Client Consistency Loss** Consistency regularization (Xie et al., 2019; Sohn et al., 2020; Berthelot et al., 2019b;a) is one of most popular approaches to learn from unlabeled examples in a semi-supervised learning setting. Conventional consistency-regularization methods enforce the predictions from the augmented examples and original (or weakly augmented) instances to output the same class label, $||p_\theta(\mathbf{y}|\pi(\mathbf{u})) - p_\theta(\mathbf{y}|\pi'(\mathbf{u}))||_2^2$, where $\pi(\cdot)$ and $\pi'(\cdot)$ are stochastic transformation functions (e.g. random data augmentations). Based on the assumption that class semantics are unaffected by small input perturbations, these methods basically ensures consistency of the prediction across the multiple perturbations of *same input*. For our federated semi-supervised learning method, we additionally propose a novel consistency loss that regularizes the *models* learned at multiple clients to output the same prediction. This novel consistency loss for FSSL, which we refer to as *inter-client consistency* loss, is defined as follows:

$$\sum_{j=1}^{H} \mathrm{KL}[p^*_{\theta^{h_j}}(\mathbf{y}|\mathbf{u}) || p_{\theta^l}(\mathbf{y}|\mathbf{u})] \quad (1)$$

where $p^*_{\theta^h}(y|x)$ is a helper agent selected from the server based on reliability, and it is not trained at the client (* denotes that we freeze the parameters). The server selects and broadcasts $H$ helper agents at each communication round. We also use data-level consistency regularization at each local client similarly to FixMatch (Sohn et al., 2020). Our final consistency regularization term $\Phi(\cdot)$ can be written as




(a) Standard Scenario (b) Disjoint Scenario

*Figure 3.* **Frameworks for FedMatch** (a) **Standard Scenario**: Active local model $l_a$ at the current communication round learns both $\sigma^{l_a}$ and $\psi^{l_a}$ on labeled and unlabeled data, respectively. Once the clients update their learned knowledge to the server, server aggregates $\sigma^{l_{1:A}}$ and $\psi^{l_{1:A}}$ through reliability-based aggregation $\Gamma(\cdot)$, while selecting the top-$H$ $\psi^{h_{1:H}}$ by their reliability. Then, the server broadcasts the aggregated $\sigma$ and $\psi$, as well as the $H$ selected $\psi^{h_{1:H}}$ to next available clients ($H$=2). (b) **Disjoint Scenario**: The global model $G$ learns $\sigma^G$ on labeled data at server and the active local clients $l_{1:A}$ at the current communication round learn $\psi^{l_{1:A}}$ on unlabeled data. Once clients update their $\psi^{l_{1:A}}$ to the server, server selects the top-$H$ most reliable $\psi^{h_{1:H}}$ by evaluating it on the validation set. Then, server broadcasts its learned $\sigma_G$ as well as the aggregated $\psi$ and top-$H$ reliable $\psi^{h_{1:H}}$ to the next available clients ($H$=2).

follows:

$$\Phi(\cdot) = \mathrm{CE}(\hat{\mathbf{y}}||p_{\theta^l}(\mathbf{y}|\pi(\mathbf{u}))) + \sum_{j=1}^{H} \mathrm{KL}[p^*_{\theta^{h_j}}(\mathbf{y}|\mathbf{u})||p_{\theta^l}(\mathbf{y}|\mathbf{u})] \tag{2}$$

where $\pi(\mathbf{u})$ performs RandAugment (Cubuk et al., 2019) on unlabeld instance $\mathbf{u}$, and $\hat{\mathbf{y}}$ is the *agreement-based* pseudo label,

$$\hat{\mathbf{y}} = \mathrm{Max}(\mathbb{1}(p^*_{\theta^l}(\mathbf{y}|\mathbf{u})) + \sum_{j=1}^{H} \mathbb{1}(p^*_{\theta^{h_j}}(\mathbf{y}|\mathbf{u}))) \tag{3}$$

where $\mathbb{1}(\cdot)$ produces one-hot labels with given softmax values , and $\mathrm{Max}(\cdot)$ outputs one-hot labels on the class that has the maximum agreements. We discard instances with low-confident predictions below confidence threshold $\tau$ when generating pseudo-labels, as done in (Sohn et al., 2020).

**Parameter Decomposition for Disjoint Learning** In the standard semi-supervised learning, learning on labeled and unlabeled data is simultaneously done with a shared set of parameters. However, this may result in the model to forget about what it learned with labeled data (see Figure 4 (c)). To tackle this, we decompose our model parameters $\theta$ into two variables, $\sigma$ for supervised learning and $\psi$ for unsupervised learning, such that $\theta = \sigma + \psi$. We perform standard supervised learning on $\sigma$, while keeping $\psi$ fixed during training, by minimizing the loss term as follows:

$$\text{minimize } \mathcal{L}_s(\sigma) = \lambda_s \mathrm{CE}(\mathbf{y}, p_{\sigma+\psi^*}(\mathbf{y}|\tilde{\pi}(\mathbf{x}))) \tag{4}$$

where $\mathbf{x}$ and $\mathbf{y}$ are from labeled set $\mathcal{S}$, and $\tilde{\pi}(\cdot)$ is a simple flip-and-shift augmentation as the same as FixMatch's (Sohn et al., 2020). For learning on unlabeled data, we perform unsupervised learning conversely on $\psi$, while keeping $\sigma$ fixed for the learning phase, by minimizing the consistency loss terms as follows:

$$\begin{aligned}\text{minimize } \mathcal{L}_u(\psi) = \lambda_u \Phi_{\sigma^*+\psi}(\cdot) + \lambda_{L_2}||\sigma^* - \psi||_2^2 \\ +\lambda_{L_1}||\psi||_1\end{aligned} \tag{5}$$

where $\lambda$s are hyper-parameters to control the learning ratio between the terms. We additionally add $L_2$- and $L_1$-Regularization on $\psi$ so that $\psi$ is sparse, while not drifting far from the knowledge that $\sigma$ has learned. This sparse parameters also enable efficient communications between clients and server.

**Reliability-based Aggregation** Since not all local models may be equally reliable as they learn on unlabeled data, evaluating the reliability of the locally learned knowledge is crucial. Therefore, we propose a *reliability-based aggregation* $\Gamma(\cdot)$ to enhance the effect of reliable knowledge, while minimizing the negative effect of the unreliable knowledge as follows:

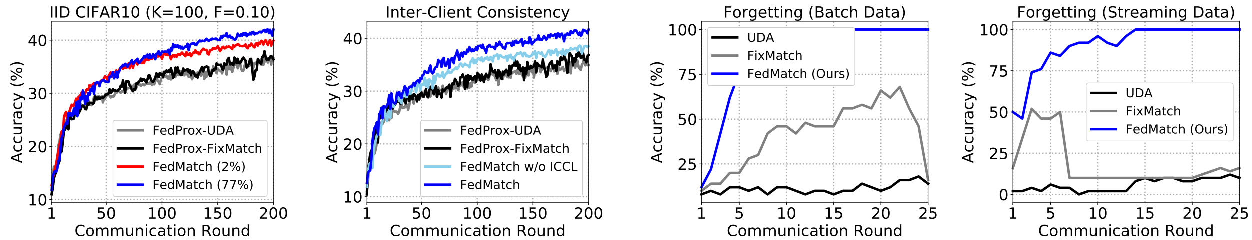
$$\Gamma(\theta^{l_{1:A}}) = \frac{\mathrm{Acc}^{l_a}}{\mathrm{TotalAcc}^{l_{1:A}}} \sum_{a=1}^{A} \theta^{l_a} \tag{6}$$

where $\mathrm{Acc}^{l_i}$ denotes the scores of local model $l_a$ on the validation set at server, and $\mathrm{TotalAcc}^{l_{1:A}}$ is total sum of all scores on $A$ number of available clients at each communication round as described in Figure 3 (a).

## 3. Experiments

We now validate our *FedMatch* on three datasets: streaming Non-IID dataset under standard scenario, and streaming non-IID dataset under disjoint scenario, and Batch IID dataset.

**Datasets 1) Streaming Non-IID Dataset:** We evaluate FedMatch on non-IID, streaming setting based on the realistic assumption for federated learning where each model




(a) $F$=0.1 (b) Inter-Client Consistency (c) Forgetting on Labeled Data

*Figure 4.* **(a) Test Accuracy Curves on Batch IID Dataset** with 100 clients ($F$=0.1) corresponding to Table 2. **(b) Effect on Inter-Client Consistency Loss** over Batch IID dataset in Standard Scenario. **(c) Forgetting on Labeled Data** in batch & streaming scenario in local SSL models. Each model performs SSL with 5 labels per class (50 in total).

*Table 1.* **Performance Comparison on Streaming Non-IID Dataset (Fashion-MNIST)** with 10 clients ($F$=1.0)

| Streaming Non-IID Dataset (Fashion-MNIST) | | | | |
|---|---|---|---|---|
| | *Standard Scenario* | | *Disjoint Scenario* | |
| **Methods** | **Acc.(%)** | **Cost** | **Acc.(%)** | **Cost** |
| Loc.SL | 61.57 | N/A | N/A | N/A |
| Loc.UDA | 50.86 | N/A | N/A | N/A |
| Loc.FixMat | 53.55 | N/A | N/A | N/A |
| F.Avg-SL | 63.75 | 100 % | 66.68 | 100 % |
| F.Prx-SL | 64.46 | 100 % | 67.05 | 100 % |
| F.Avg-UDA | 52.10 | 100 % | 46.53 | 100 % |
| F.Prx-UDA | 52.55 | 100 % | 45.90 | 100 % |
| F.Avg-FixMat | 56.31 | 100 % | 50.19 | 100 % |
| F.Prx-FixMat | 54.69 | 100 % | 52.51 | 100 % |
| **FedMatch-S** | **61.34** | **102 %** | **58.64** | **60 %** |
| **FedMatch-D** | **63.61** | **177 %** | **59.40** | **100 %** |

works with locally-generated private data. Specifically, we intentionally control the distribution of the number of instances per class for each client to simulate such biased environments. We use Fashion-MNIST dataset for this setting, and split Fashion-MNIST ($70,000$) into training ($63,000$), valid ($3,500$), and test ($3,500$) sets. For the standard scenario, we extract 5 labeled instances per class ($C$=5) for each client ($K$=10) from train set, while extracting 50 instances per class once for a labeled set $\mathcal{S}^G$ (500 for both scenarios) at server (disjoint scenario). We discard labels for the rest of instances to construct an unlabeled set $\mathcal{U}$ ($62,000$). Then, we split $\mathcal{U}$ into $\mathcal{U}^{l_{1:100}}$ based on a class-wise non-iid distribution. For individual local data $\mathcal{U}^{l_k}$, we again split all instances into $\mathcal{U}_t^{l_k}$, $t \in \{1, 2, ..., T\}$, where $T$ is the number of total streaming steps (we set $T$=10). **2) Batch IID Dataset**: We also validate our models on an IID dataset constructed out of CIFAR-10 for the standard scenario. We split CIFAR-10 ($60,000$) into training ($54,000$), valid ($3,000$), and test ($3,000$) sets. With the training set, we extract 5 labeled instances per class ($C$=10) for each client ($K$=100) as labeled datasets. We remove labels for the rest of instances to use them as the unlabeled set $\mathcal{U}(49,000)$. Then, we evenly split $\mathcal{U}$ into $\mathcal{U}^{l_{1:100}}$ and distribute them across 100 clients, such that local models $l_{1:100}$ learn on corresponding $\mathcal{S}^{l_{1:100}}$ and $\mathcal{U}^{l_{1:100}}$ during training.

*Table 2.* **Performance Comparison on Batch IID Dataset (CIFAR-10)** with 100 clients ($F$=[0.05, 0.1]) during 200 rounds.

| Batch IID Datset (CIFAR-10) with 100 Clients | | | | |
|---|---|---|---|---|
| | $F$=0.05 | | $F$=0.10 | |
| **Methods** | **Acc.(%)** | **Cost** | **Acc.(%)** | **Cost** |
| F.Avg-SL | 47.23 | 100 % | 47.87 | 100 % |
| F.Prx-SL | 47.54 | 100 % | 48.01 | 100 % |
| F.Avg-UDA | 35.27 | 100 % | 35.20 | 100 % |
| F.Prx-UDA | 34.93 | 100 % | 36.67 | 100 % |
| F.Avg-FixMat. | 32.33 | 100 % | 36.27 | 100 % |
| F.Prx-FixMat. | 36.83 | 100 % | 36.37 | 100 % |
| **FedMatch-S** | **38.43** | **102 %** | **38.83** | **102 %** |
| **FedMatch-D** | **41.67** | **177 %** | **41.97** | **177 %** |

**Baselines and Experimental Setup** **(1) Local-SL**: local Supervised Learning with full labels. **(2)-(3) Local-UDA/FixMatch**: local semi-supervised learning baselines, without sharing knowledge. **(4)-(5) FedAVG/Prox-SL**: supervised learning with full labels while sharing local knowledge via FedAvg/Prox frameworks. **(6)-(7) FedAvg/Prox-UDA**: naive combination of FedAvg/Prox with UDA. **(8)-(9) FedAvg/Prox-FixMatch**: naive combination of with FixMatch with FedAvg/Prox. We use a modified AlexNet-like networks (Serra et al., 2018) as the backbone networks for all methods and we use SGD with momentum 0.9 and adaptive-learning rate decay introduced in (Serra et al., 2018) with the initial learning rate is $1e{-}4$. We implement Training Signal Annealing (TSA) for UDA and we set $\lambda_u$=1 for both UDA and FixMatch, as reported. The confidence level is set to 0.75 for FixMatch and our model. For most of experiments, we set $\lambda_u$=1, $\lambda_s$=10, $\lambda_{L_2}$=10, $\lambda_{L_1}$=[0 : 0.01] for our method.

### 3.1. Experimental Results

**Results on Streaming Non-IID Dataset** We perform experiments under both standard and disjoint scenarios, utilizing 10 clients with fraction of connection ($F$=1.0) during 10 rounds per streaming steps ($T$=10). We set the batch size of the labeled set ($B^S$=10) and the unlabeled set ($B^U$=50) differently. We set number of epoch $E$ to 1 per round. Table 1 shows the results on these experiments. We observe that while naively combining federated learning with semi-supervised learning results in mild improvement in the performance (1.69%p with UDA and 4.21%p with FixMatch), our FedMatch variants significantly outperform all of them by large margins on both scenarios. Specifically, FedMatch-Dense obtains 7.3%p performance gain over the best performing baseline, FedAvg-FixMatch in the standard scenario, and obtains 6.89%p improvement over the best basline, FedProx-FixMatch in the disjoint scenario. Surprisingly, FedMatch obtains comparable performance to supervised learning methods which have $100\%$ of the data labeled (FSSL methods have labels on only $10\%$ of the data). Moreover, FedMatch-Sparse (FedMatch-S) obtains marginally lower performance over FedMatch-Dense (FedMatch-D), but it is more efficient in terms of memory and communication cost. Also, it requires the lowest communication cost for the disjoint scenario.

**Results on Batch IID Dataset** We further validate our models on IID dataset for the standard scenario (see Table 2). We set the same setting as the above experiment, except $F$=$[0.05, 0.1]$, $R$=200. We use 5 ground truth instances per class (for each client) for all base models, except for supervised learning (SL) models that use full labels. We visualize the test accuracy curves for our models ($F$=0.1) and naive FedAvg-SSL in Fig. 4 (a). Our method, FixMatch-D (Blue line), trains learns and consistently outperforms the naive federated semi-supervised learning frameworks (FedProx-UDA/FixMatch) that show similar performance during training. Table 2 shows performance for all base models. Our models significantly outperforms naive Fed-SSL methods with $1.6\%$p - $9.3\%$p higher accuracy.

**Ablation Study** In Figure 4 (b), we experiment on the effectiveness of our inter-client consistency loss on Batch IID dataset with $100$ clients ($F$=0.05). According to the figure, we observe that the performance has slightly dropped without inter-client consistency loss, which the gap is clear evidence that our method effectively utilizes reliable knowledge from other clients. Moreover, our model without inter-client consistency loss still outperforms base models (FedProx-UDA/FixMatch). This additionally implies that our proposed parameter decomposition method has also meaningful effects. As shown in Figure 4 (c), our method successfully preserves learned knowledge from labeled data. We perform SSL with only 5 labels per class with $1,000$ unlabeled instances in both streaming (5 rounds per streaming step) and batch settings, and we measure forgetting on labeled set at each training steps. As shown, preserving reliable knowledge from labeled data leads to performance improvement of our proposed model without inter-client consistency loss over naive FSSL models.

# 4. Related Work

**Federated Learning**: Federated Learning collaboratively learns a global model while communicating with multiple clients that train on their own private local data. A variety of approaches for averaging local weights at server have been introduced in the past few years. FedAvg (McMahan et al., 2017) performs weighted-averaging on local weights according to the local train size. FedProx (Li et al., 2018) uniformly averages the local updates while clients perform proximal regularization against the global weights, while FedMA (Wang et al., 2020) matches the hidden elements with similar feature extraction signatures in layer-wise manner when averaging local weights. **Semi-Supervised Learning**: Semi-Supervised Learning (SSL) is the problem of learning with both labeled and unlabeled data. While there exist numerous work on SSL, we mainly discuss consistency regularization approaches. Consistency regularization techniques(Rasmus et al., 2015; Sajjadi et al., 2016) assume that the class semantics will not be affected by transformations of the input instances, and enforce the model output to be the same across different input perturbations. Some extensions to this technique perturb inputs adversarially (Miyato et al., 2018), through dropout (Srivastava et al., 2014), or through data augmentation (French et al., 2017). UDA (Xie et al., 2019) and ReMixMatch (Berthelot et al., 2019a) use two sets of augmentations, weak and strong, and enforce consistency between the weakly and strongly augmented examples. Recently, FixMatch (Sohn et al., 2020) uses pseudo-labeling in addition to enforcing consistency between weak-strong augmented pairs.

# 5. Conclusion

In this work, we proposed a novel problem of *Federated Semi-Supervised Learning* (FSSL) where each client learns with only partly labeled data (standard scenario), or work with completely unlabeled data with supervised labels only available at the server (disjoint scenario). To tackle this problem, we propose a novel method, *Federated Matching* (FedMatch), which introduces the *Inter-Client Consistency Loss* that aims to maximize the agreement between the models trained at different clients, and *Additive Parameter Decomposition* which decomposes the parameters into one for labeled data and the other for unlabeled data to prevent forgetting of the knowledge learned on labeled data. Through extensive experimental validation, we show that FedMatch significantly outperforms both local semi-supervised learning methods and naive combinations of federated learning algorithms with semi-supervised learning on diverse and realistic scenarios.

## References

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019b.

Yang Chen, Xiaoyan Sun, and Yaochu Jin. Communication-efficient federated deep learning with asynchronous model update and temporally weighted aggregation. *arXiv preprint arXiv:1903.07424*, 2019a.

Yujing Chen, Yue Ning, and Huzefa Rangwala. Asynchronous online federated learning for edge devices. *arXiv preprint arXiv:1911.02134*, 2019b.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2(4): 7, 2019.

Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

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

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

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

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. *arXiv preprint arXiv:1801.01423*, 2018.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

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

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. In *arXiv preprint arXiv:1904.12848*, 2019.

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.