Improving Semi-supervised Federated Learning by Reducing the Gradient Diversity of Models

Zhengming Zhang[1*] ,Yaoqing Yang[2*], Zhewei Yao[2*], Yujun Yan[3],
Joseph E. Gonzalez[2], Kannan Ramchandran[2], Michael W. Mahoney[2,4]
[1] School of Information Science and Engineering, Southeast University
[2] Department of EECS, University of California, Berkeley
[3] Department of EECS, University of Michigan, Ann Arbor
[4] International Computer Science Institute
Email: zmzhang@seu.edu.cn,yujunyan@umich.edu
{yqyang, zheweiy, jegonzal, kannanr, mmahoney}@berkeley.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671693

***Abstract*—Federated learning (FL) is a promising way to use the computing power of mobile devices while maintaining the privacy of users. Current work in FL, however, makes the unrealistic assumption that the users have ground-truth labels on their devices, while also assuming that the server has neither data nor labels. In this work, we consider the more realistic scenario where the users have only unlabeled data, while the server has some labeled data, and where the amount of labeled data is smaller than the amount of unlabeled data. We call this learning problem semi-supervised federated learning (SSFL). For SSFL, we demonstrate that a critical issue that affects the test accuracy is the large *gradient diversity* of the models from different users. Based on this, we investigate several design choices. First, we find that the so-called *consistency regularization loss* (CRL), which is widely used in semi-supervised learning, performs reasonably well but has large gradient diversity. Second, we find that Batch Normalization (BN) increases gradient diversity. Replacing BN with the recently-proposed Group Normalization (GN) can reduce gradient diversity and improve test accuracy. Third, we show that CRL combined with GN still has a large gradient diversity when the number of users is large. Based on these results, we propose a novel grouping-based model averaging method to replace the FedAvg averaging method. Overall, our grouping-based averaging, combined with GN and CRL, achieves better test accuracy than not just a contemporary paper on SSFL in the same settings (>10%), but also four supervised FL algorithms.**

***Index Terms*—federated Learning, semi-supervised learning, gradient diversity**

I. Introduction

State-of-the-art machine learning models can benefit from the large amount of user data privately held on mobile devices, as well as the computing power locally available on these devices. In response to this, federated learning (FL) has been proposed [1], [2]. In a typical FL pipeline, a server and some users jointly learn a model in multiple rounds. In each round, models are updated locally (e.g., on users' devices) based on private user data, the server aggregates the updated models sent from the users, and the server then shares the aggregated model with the users for the next round.

In FL, it is commonly assumed that the data stored on the local devices are fully annotated with ground-truth labels, and that the server does not have any labeled data [1]–[3]. However, this assumption does not hold in practice. On the one hand, there is not a sufficient supply of labeled data on the users' side [4], as labeling data requires both time and domain knowledge [5], [6]. On the other hand, the server, which is often hosted by organizations, is more likely than a single user to acquire labeled data. To give some concrete examples, consider two scenarios: cross-device FL (in which users are mobile devices) and cross-silo FL (in which users are organizations) [7]. In a cross-device scenario, where a central server trains an object detector on images with the help of mobile users, the server can use a public dataset, e.g., [8], to obtain labels, while the users often do not have images with ground-truth bounding boxes. In a cross-silo scenario, where multiple medical institutes work together to diagnose a disease, the disease may be newly discovered by one medical institute, and so no labeled samples are present at other institutes [4]. In these scenarios, the typical supervised FL setting is not appropriate.

Motivated by these practical scenarios, we study the *semi-supervised federated learning* (SSFL) setting. In SSFL, users only have access to unlabeled data, while the server only has a small amount of labeled data. In addition to this main setup (in which users have unlabeled data only), we also compare our method to the state-of-the-art in another "label-at-client" scenario [9], in which users have a limited amount of labeled data, while the server does not have data. Our method outperforms [9] in both the main setup and the "label-at-client" scenario by a large margin (>10%) in exactly the same setting, e.g., we both use ResNet-9. By considering different scenarios in SSFL, the goal is to train a model that can utilize both

There is some nuance here that the medical institute with ground-truth labels is physically different from the server, and the server itself does not have data. However, it will become apparent that this nuance does not affect the mathematical formulation considered in our paper because we can assume (virtually) that the server and the institute with labels are co-located and work together as a new server.

labeled and unlabeled data for different situations. In this context, our main contributions are the following.

1) **Demonstrating the importance of "gradient diversity."** We demonstrate the importance of reducing the gradient diversity [10], a notion which captures the dissimilarity between local gradient updates of users, in SSFL. First, we show that the consistency regularization loss (CRL) [11] can achieve reasonably good test accuracy, but it still has significantly larger gradient diversity than supervised FL. Then, we show that replacing the batch normalization (BN) [12] in the model with group normalization (GN) [13] can reduce gradient diversity and enhance test accuracy in the SSFL setting. Finally, we propose a grouping-based model averaging technique to replace FedAvg [2], to reduce gradient diversity further and to increase accuracy, especially when there are a large number of users.
2) **Proposing a strong baseline.** By proposing solutions to reduce gradient diversity, we obtain a strong SSFL approach. Our method outperforms another SSFL approach from a contemporary paper [9] in the same settings by 14.79%-18.10% in test accuracy. Our method also achieves comparable or better accuracy than four existing *supervised* FL approaches that do not use GN or the grouping-based averaging. Specifically, our approach is 0.80%/0.29% better than EASGD/OverlapSGD [14], [15], despite having a lower communication frequency, and our approach is 14.44%/11.14% better than FedAvg/DataSharing [2], [16], even when the degree of our non-iidness (in the sense of different distributions of classes at different users) is higher.
3) **Extensive empirical evaluation.** We evaluate the proposed solution by varying different environmental factors and testing on multiple datasets. The environmental factors include different levels of non-iidness, the *communication period* (i.e., the number of local update steps at each user between two communication rounds), the total amount of labeled data in the server, the number of users, and the number of users that communicate with the server in each communication round. Interestingly, through extensive empirical evaluation, we discover that large gradient diversity can arise from a large number of communicating users (see § III-E), which leads to the grouping-based averaging method.

Overall, by formulating the SSFL problem, analyzing the key limitation of large gradient diversity, selecting different design choices to reduce the gradient diversity, and thoroughly evaluating our design under different environmental factors, we provide a strong baseline for this SSFL setting. This strong baseline can achieve comparable or better accuracy than the state-of-the-art methods in both semi-supervised and supervised FL. The proposed method also only focuses on a few crucial components (e.g., normalization) that are easy to change in practice.

II. Semi-supervised federated learning

A. Basic setup

In this subsection, we discuss the basic setup of SSFL. There exist a cloud server and K users/devices. Similar to the common FL setup [1], the users and the server collaborate to train a model in multiple rounds by exchanging and updating model weights. For each round of communication, we allow the number of participating users connected to the server, which we denote as C, to be smaller than K, as is done commonly [7]. This is because, for example, some mobile devices only participate in the learning when being charged [7]. Assuming $C \leq K$ for each round of communication can simulate this drop-and-reconnect case.

We denote the labeled dataset at the server as $D_s = \{(x_i, y_i)\}_{i=1}^{N_s}$, and the unlabeled dataset stored at the k-th user as $D_k = \{x_i\}_{i=1}^{N_k}$, for $k \in \{1, \ldots, K\}$. Here, N_s (N_k) is the number of labeled (unlabeled) samples available at the server (k-th user). Also, similar to the standard FL setup, no raw data are exchanged between the server and the users. That is to say, the server can only use the dataset D_s, and the k-th user can only use the local dataset D_k. Note that the data distributions at different users are non-iid [16], [17]. In this work, we consider image classification as a representative SSFL task.

We now describe the SSFL training pipeline which can be slightly different from the standard FL setup. Denote the local weights at the k-th user as w_k. Since the server has its own dataset D_s, unlike the standard FL setup, it also updates its own weights w_s. Denote the averaged weights at the server as w_{avg} (which is different from w_s). At round t, the server sends the averaged model weights w_{avg}^t to the users. Each user, upon receiving w_{avg}^t, locally updates its own model weights to w_k^t and transmits w_k^t to the server. At the same time, the server also has to update its own model weights from w_{avg}^t to w_s^t using the labeled dataset D_s. Then, the server computes an averaged model w_{avg}^{t+1} using all the received models, including its own model w_s^t. Finally, it proceeds to the next round and sends w_{avg}^{t+1} to the users. Our basic SSFL setup is illustrated in Fig. 1.

B. Gradient diversity, and the ways to reduce it

In this subsection, we present the definition of gradient diversity from [10]. Then, we motivate several design choices to reduce gradient diversity in SSFL. As we have discussed in the introduction, reducing the gradient diversity value is crucial for SSFL.

Definition 1 (Metric for gradient diversity)**.** *The gradient diversity is defined as:*

$$\Delta^t(w) = \sum\nolimits_{k \in \mathcal{C}_t} \left\| \nabla w_k^t \right\|_2^2 / \left\| \sum\nolimits_{k \in \mathcal{C}_t} \nabla w_k^t \right\|_2^2, \tag{1}$$

where $\mathcal{C}_t$ denotes the set of participating users at round t, w_k^t represents the model weights held by the k-th user at the beginning of round t, and ∇w_k^t represents the gradient of w_k^t evaluated on all data held by the k-th user.

Gradient diversity [10] measures the dissimilarity between the local gradient updates of users. In SSFL, when gradient diversity is too large, the weights from different users are updated towards "different directions," and it is thus problematic

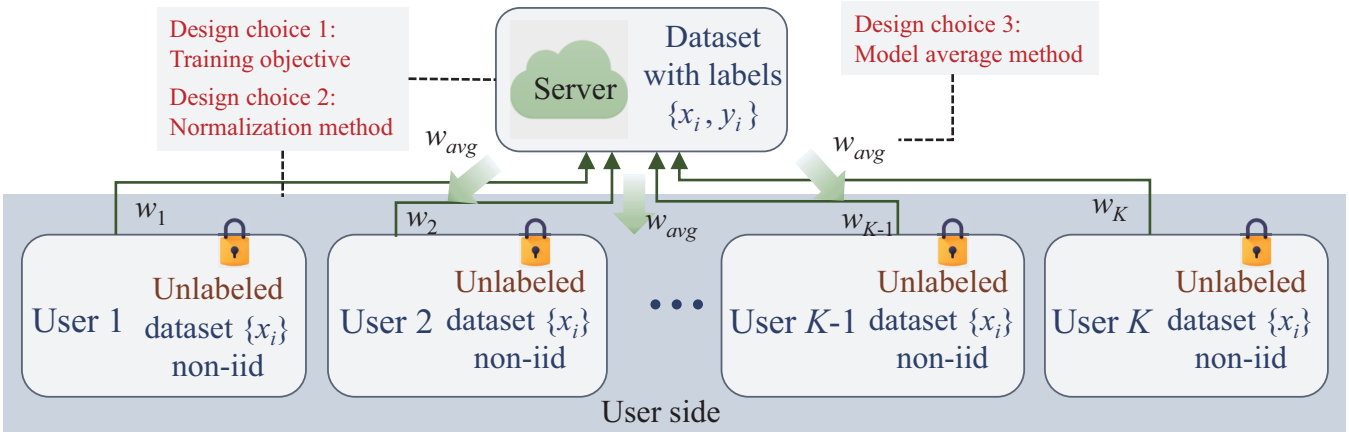

Fig. 1: *Semi-supervised federated learning (*SSFL*). Only the server has access to labeled data, i.e., the data stored in local users are unlabeled. Furthermore, the data distributions across different users are non-iid.*

to directly average them, as is done in the common model averaging method known as FedAvg [2]. Compared to [10], we pay more attention to empirical evaluations, and we aim to provide easy-to-implement solutions in the semi-supervised setting using methods motivated by gradient diversity. When calculating gradient diversity, we only include the users in $\mathcal{C}_t$, i.e., those who participate in this particular communication round, because only these weights are averaged. We provide a thorough ablation study on sixteen different ways of calculating gradient diversity/dissimilarity in § III-E.

In what follows, we motivate three design choices, which are also shown in Fig. 1, that can affect gradient diversity.

1) **Training objective.** Since there are no labels on the users' side, we have to choose appropriate loss functions carefully when updating local models at the users' side.
2) **Normalization.** Normalization (e.g., BN) has become standard in deep neural network models. The SSFL setting requires a careful choice of specific normalization methods because both non-iidness and the lack of labels can affect the normalization coefficients.
3) **Model averaging.** The way the server computes the aggregated model from the models that it receives is also an important design choice. We only consider ways to average the model weights, i.e., we do not consider model ensembling or distillation techniques, which can be time-consuming in multiple rounds [18].

C. Environmental factors for evaluation

In this subsection, we list some environmental factors that can affect the test accuracy of SSFL algorithms. These factors are not controllable by the designer, and they are independent of the design choices listed in § II-B. However, these factors are helpful to evaluate different design choices, and they can potentially display the weakness of certain solutions. The following factors are considered.

1) **Non-iidness** R: the metric that we use to measure the non-iidness of our data; see Definition 2.
2) **Communication period** T: during two consecutive communications, the number of gradient update steps locally done by the users and the server.
3) **Server data number** N_s: the number of labeled data in the server.
4) **User number** K: the total number of users.
5) **Number of participating users** C: during each communication round, the group of users who send their models to the server.

Varying non-iidness to evaluate our solution. Among the five environmental factors listed above, evaluating with different non-iidness requires special care, because we have to change the dataset to get different degrees of non-iidness. Here, we follow convention and evaluate using synthesized non-iid datasets that have different *class distribution skews* [15]–[17], [19], [20], e.g., a single user can have more data for one class or a couple of classes than others.

To quantify the class distribution skew in our experiments, we use the average total variation distance in Definition 2. In the definition, the *empirical* class distribution of the data D_k at the k-th user is denoted by $P_k \in \mathbb{R}^d$, where d is the number of classes. Clearly, $\sum_{j=1}^{d} P_k[j] = 1$, for all $1 \leq k \leq K$. Recall that K is the number of users/devices.

Definition 2 (Metric R for non-iid level)**.** *The non-iid metric R to measure the class distribution skew is defined as:*

$$R = \frac{1}{K(K-1)/2} \sum\nolimits_{1 \leq k < m \leq K} \|P_k - P_m\|_1/2, \qquad (2)$$

where $\|\cdot\|_1$ is the L_1 norm.

Here $\|P_k - P_m\|_1/2$ is the (normalized) total variation distance, which takes value in $[0, 1]$, and $K(K-1)/2$ is the number of user pairs, i.e., it is the mean total variation distance averaged over pairs of users. In particular, $0 \leq R \leq 1$ [21]. When data are distributed in such a way that each user has the same empirical class distribution which is uniform $P_k = [1/d, ..., 1/d], \forall k$, we have $R = 0$; and in another extreme, when $K = d$ and each user only has samples from one class, we have $R = 1$.

Remark 1 (Different data sizes)**.** *The metric R in Definition 2 does not explicitly consider the effect of different data sizes N_k's at different users. We focus on the case when N_k's are equal to each other, while slight difference may arise when the overall number of samples is not divisible by the number of users. Notice that, we consider similar data sizes at different users, but we do not restrict ourselves to the case of uniform class distribution. Specifically, we have tested on datasets with non-uniform class sizes, e.g., SVHN dataset.*

We synthesize datasets with a specific R value in $[0, 1]$ to evaluate our SSFL algorithm with different non-iidness. The specific data synthesis and distribution procedures to achieve a specific non-iid value R are not crucial and are not included in the paper due to space limitation. One can find the procedures in Appendix A.2, page 17 in our full version [22] online.

III. Design choices driven by gradient diversity

In this section, we study the three SSFL design choices discussed in § II-B. We first present the details of the three choices. Then, we show how they can reduce gradient diversity.

A. Design choice 1: training objective

In this subsection, we present the training objective and focus on an existing semi-supervised loss called consistency regularization loss (CRL) [9], [11], [23], [24], which has been standard for deep semi-supervised learning.

In particular, the server loss L_s and the user loss L_k (of the k-th user) are defined as follows:

$$L_s = \frac{1}{N_s} \sum\nolimits_{(x_i, y_i) \in D_s} l\left(y_i, f_s(\alpha(x_i); w_s)\right), \tag{3}$$

$$L_k = \frac{1}{N_k} \sum_{x_i \in D_k} \mathbf{1}_{\max(\bar{y}_i) \geq \tau} l\left(\arg\max(\bar{y}_i), f_k(A(x_i); w_k)\right), \tag{4}$$

where (1) D_s is the set of N_s labeled samples, (2) D_k is the set of N_k unlabeled samples owned by the k-th user, (3) w_s (w_k) are the weights of server model f_s (k-th user model f_k), (4) $l(\cdot, \cdot)$ is the cross-entropy loss, $\alpha(\cdot)$ and $A(\cdot)$ are two data augmentation functions which we will soon describe in Remark 1, (5) $\bar{y}_i = f_k(\alpha(x_i); w_k)$ is the prediction of the model f_k on the augmented sample $\alpha(x_i)$, (6) $\mathbf{1}$ is the indicator function, (7) and τ is the threshold hyperparameter which helps decide which samples have high confidence to be trained, i.e., the term $\mathbf{1}_{\max(\bar{y}_i) \geq \tau}$. We refer to training with Eq. 3 and Eq. 4 as the *CRL training objective*.

Remark 1 (Data augmentation)**.** *We now discuss the data augmentations in Eq. 3 and Eq. 4. In [11], the authors use two different types of data augmentations (DA): the standard flip-and-shift augmentation $\alpha(\cdot)$ (referred to as* weak DA*); and the RandAugment [25] $A(\cdot)$ (referred to as* strong DA*). Here, the latter RandAugment uses two different augmentation methods (i.e., shift and crop) out of twelve possible augmentation methods (e.g., rotate, shift, solarize, etc.) for one image. We refer the interested readers to [25] for a detailed explanation. The key idea behind using two DAs (i.e., weak DA and strong DA) is that the predictions of the same image with two data augmentations should be similar to each other. Recall that, on the user side, the data have no labels. Therefore, using this approach, we can use the pseudo-labels generated from weak DA samples to supervise strong DA samples, which is the loss between $\arg\max(\bar{y}_i)$ and $f_k(A(x_i); w_k)$ in Eq. 4. This is shown in [11] to boost the testing performance.*

Other training objectives. To study the CRL training objective, we compare it to two other training objectives. One uses classical self-training similar to the way defined in [26], which is also called "pseudo-labeling" in [11]:

$$L_k = \frac{1}{N_k} \sum_{x_i \in D_k} \mathbf{1}_{\max(\bar{y}_i) \geq \tau} l\left(\arg\max(\bar{y}_i), f_k(\alpha(x_i); w_k)\right). \tag{5}$$

This loss can be explained as replacing two augmentations $\alpha(\cdot)$ and $A(\cdot)$ in the CRL training objective Eq. 4 with a single standard flip-and-shift augmentation $\alpha(\cdot)$. It is called self-training because the pseudo-labels obtained by applying $\arg\max$ to the model's output $\bar{y}_i = f_k(\alpha(x_i); w_k)$ are used to supervise the model's output $f_k(\alpha(x_i); w_k)$ itself. We refer to Eq. 5 as the *self-training objective*.

The other training objective assumes that the users have (oracle) ground-truth labels, and it uses standard empirical risk minimization for the user loss, e.g., used in [2]:

$$L_k = \frac{1}{N_k} \sum\nolimits_{x_i \in D_k} l\left(y_i, f_k(\alpha(x_i); w_k)\right), \tag{6}$$

where y_i is the (oracle) ground-truth label of x_i. We refer to Eq. 6 as the *supervised training objective*.

B. Design choice 2: normalization method

In this subsection, we describe the next design choice regarding the normalization method. Recent papers [27], [28] find that in supervised FL, the performance of group normalization (GN) is usually much better than that of batch normalization (BN). In contrast to BN, which normalizes the feature maps over the batch, height, and width dimensions, GN normalizes the feature maps over the channel, height, and width dimensions. We conjecture that the improvement of applying GN in FL is due to the reduced gradient diversity, and we thus empirically evaluate the effects of these two different normalization methods.

C. Design choice 3: model averaging

In this subsection, we study model averaging methods. We focus on a novel grouping-based averaging method. The main idea is to divide the C communication users in each round into $S > 1$ groups and then perform the average group-wise. Specifically, after collecting all C model weights from the communication users, the server randomly divides them into S equal-sized groups $\{G_i^t\}_{i=1}^S$, and updates the averaged weights according to:

$$\begin{cases} w_{avg,i}^{t+1} = \left(w_s^t + \sum_{k \in G_i^t} w_k^t\right) / (|G_i^t| + 1), \ \forall i \in \{1 \dots S\} \\ w_{avg}^{t+1} = \sum_{i=1}^S w_{avg,i}^{t+1} / S. \end{cases} \tag{7}$$

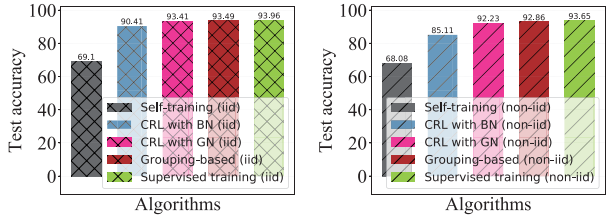

Fig. 2: *(Left) Test accuracy of different methods in the iid setting ($R = 0.0$) on Cifar-10. (Right) Test accuracy of different methods in the non-iid setting ($R = 0.4$) on Cifar-10.*

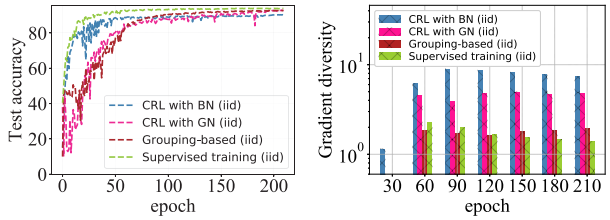

Fig. 3: *(Left) Convergence curves of different methods on Cifar-10 in the iid setting ($R = 0$). (Right) The corresponding gradient diversity during training.*

In the equation above, $w_{avg,i}^{t+1}$ represents the averaged weights in each group, and w_{avg}^{t+1} is the average of these averaged weights. After computing $w_{avg,i}^{t+1}$ and w_{avg}^{t+1}, the server broadcasts $w_{avg,i}^{t+1}$ to the user group G_i^t, and it uses w_{avg}^{t+1} for the training (updates) done by the server itself on the labeled data. It is worth noting that the groups $\{G_i^t\}_{i=1}^S$ change with t because the set of participating users $\mathcal{C}_t$ change with time. We compare the grouping-based averaging method to FedAvg:

$$w_{avg}^{t+1} \overset{FedAvg}{=} \left(w_s^t + \sum\nolimits_{k\in\mathcal{C}_t} w_k^t\right)/(C+1), \tag{8}$$

where $\mathcal{C}_t$ denotes the set of participating users with size C in each round.

D. Different SSFL *methods and gradient diversity*

In this subsection, we study five different ways to combine the three design choices:

- *CRL with BN* uses Eq. 3 and Eq. 4 as the training objective. It uses BN as the normalization method and FedAvg in Eq. 8 as the model averaging method.
- *Self-training* uses Eq. 5 as the training objective. It also uses BN and FedAvg.
- *Supervised training* uses Eq. 6 as the training objective. It also uses BN and FedAvg.
- *CRL with GN* uses Eq. 3 and Eq. 4 as the training objective. It replaces BN with GN, and it uses FedAvg.
- *Grouping-based* uses the same CRL training objective and GN, as in CRL with GN, but it uses the grouping-based averaging method in Eq. 7 instead of FedAvg.

We compare CRL with BN to self-training and supervised training to show where the CRL training objective stands compared to both semi-supervised and supervised algorithms. We compare CRL with BN to CRL with GN to show which normalization method is better. Further, we compare CRL with GN with the grouping-based method to show which model averaging method is better. The grouping-based solution combines CRL, GN, and our grouping-based averaging method. This solution is our main algorithm.

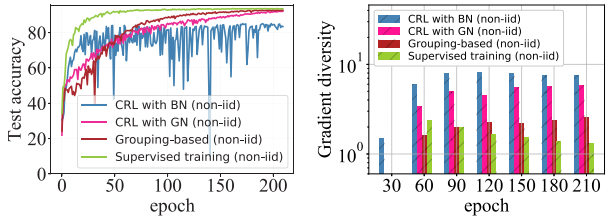

Fig. 4: *(Left) Convergence curves of different methods on Cifar-10 in the non-iid setting ($R = 0.4$). (Right) The corresponding gradient diversity during training.*

Experiment settings. In this section, we use ResNet-18 [29] on Cifar-10. For the environmental factors in § II-C, we set $T = 16$, $K = 10$, $C = 10$, and $N_s = 1000$. We compare under two R values, with $R = 0.4$ referred to as the non-iid case, and $R = 0$ referred to as the iid case. The threshold τ used in Eq. 3 and Eq. 4 is chosen to be 0.95, the same as in [11].

Results. See Fig. 2. From the test accuracy, we have the following observations.

- When restricted to either the iid or the non-iid case, CRL improves significantly over self-training, but it cannot achieve the accuracy of supervised training.
- By comparing CRL with BN to CRL with GN, we show that GN improves the test accuracy.
- By comparing CRL with GN to the grouping-based method, we show that the grouping-based averaging improves the test accuracy compared to FedAvg.

Gradient diversity analysis of different methods. Now, we use the gradient diversity in Definition 1 to analyze different design choices. See Fig. 3 and Fig. 4. The left plot shows the convergence curves. The right plot shows the gradient diversity values. We have the following observations:

- When restricted to either the iid or the non-iid case, GN reduces gradient diversity compared to BN.
- Similarly, when restricted to either the iid or the non-iid case, grouping-based averaging reduces gradient diversity compared to FedAvg (see the comparison to CRL with GN, which uses FedAvg).
- The grouping-based method has a comparable gradient diversity value to supervised training.

E. Reducing gradient diversity using grouping-based average

In this subsection, we further demonstrate the importance of reducing gradient diversity. We focus on the setting with a large number of communicating users C. This setting is particularly important because it demonstrates the weakness of the simple averaging method in FedAvg and motivates our grouping-based averaging. Thus, we present this setting before more empirical evaluations in § IV.

We consider EMNIST, which is a widely used dataset for FL. We compare FedAvg and our grouping-based method in

Table I: *Accuracy versus the number of communicating users C on EMNIST*

Dataset	$K = 47$ $C = 10$	$K = 47$ $C = 30$	$K = 47$ $C = 47$
EMNIST (FedAvg)	83.07%	79.05%	65.48%
EMNIST (Grouping-based)	84.43%	83.12%	82.95%

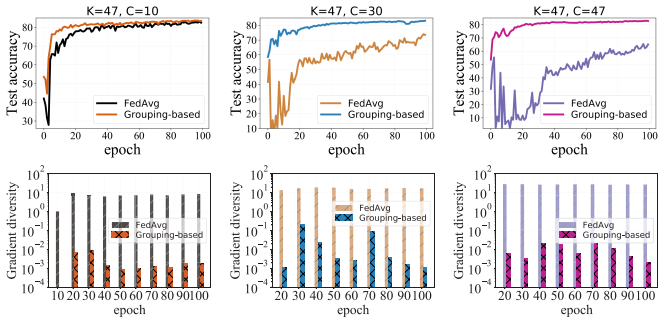

Fig. 5: *(Top) Convergence curves of FedAvg method on EMNIST when (left) $C = 10$, (middle) $C = 30$ and (right) $C = 47$. (Bottom) The corresponding results on gradient diversity.*

Tab. I. Similar to § III-D, we compare these two by letting both methods use CRL and GN. From Tab. I, one can clearly see that a large C decreases the performance significantly if one uses FedAvg. Particularly, the $K = C = 47$ case is lower than $C = 10$ by 17.59%. However, this reduction in accuracy can be mitigated if we use the grouping-based method, which is only 1.48%.

We proceed to study why the grouping-based averaging performs significantly better than FedAvg for the particular case when C is large. See Fig. 5. From the results, we can see that the gradient diversity increases significantly for a large number of communicating users. We can also see that grouping-based averaging can reduce gradient diversity and increase accuracy.

Ablation study on gradient diversity. To understand further the relationship between gradient diversity and the grouping-based averaging, we consider different ways of calculating gradient diversity.

First, we can remove the square operation in Eq. 1, and we only use the ℓ_2-norm to measure gradient diversity:

$$\Delta_1^t(w) = \sum\nolimits_{k\in\mathcal{C}_t} \left\|\nabla w_k^t\right\|_2 / \left\|\sum\nolimits_{k\in\mathcal{C}_t} \nabla w_k^t\right\|_2. \tag{9}$$

Second, we can also replace the ℓ_2-norm with the ℓ_1-norm, which leads to the following two alternatives with/without the square operation:

$$\Delta_2^t(w) = \sum\nolimits_{k\in\mathcal{C}_t} \left\|\nabla w_k^t\right\|_1^2 / \left\|\sum\nolimits_{k\in\mathcal{C}_t} \nabla w_k^t\right\|_1^2. \tag{10}$$

$$\Delta_3^t(w) = \sum\nolimits_{k\in\mathcal{C}_t} \left\|\nabla w_k^t\right\|_1 / \left\|\sum\nolimits_{k\in\mathcal{C}_t} \nabla w_k^t\right\|_1. \tag{11}$$

Third, we can change the set $\mathcal{C}_t$ in the computation. Note that in all of the definitions above, we calculated gradient diversity only using the gradients from the users. Therefore, the set $\mathcal{C}_t$ only contains users. However, we can also include the server gradient in the calculation of gradient diversity.

Fourth, we can change the way of computing each individual gradient. We notice that in FL, the local gradient updates are not aggregated directly. Instead, sequential gradient updates are applied to each user. Then, the updated weights from the users are averaged. Thus, instead of calculating the diversity of gradient ∇w_k^t evaluated on all the user data, we can define ∇w_k^t as the difference between the model before and after local gradient updates, i.e.

$$\nabla w_k^t = \begin{cases} w_k^t - w_{avg}^{t-1}, \text{for FedAvg}, \\ w_k^t - w_{avg,i}^{t-1}, \text{for grouping-based}, \end{cases} \tag{12}$$

where i is the index of group to which user k belongs; see Eq. 7. It can be seen that Eq. 12 is the cumulative change in weights after the local gradient updates. We can substitute the above-defined gradient Eq. 12 into Eq. 1 and Eq. 9-Eq. 11 to calculate gradient diversity.

Thus, we can either perform the square operation or not, either use the ℓ_2-norm or the ℓ_1-norm, either include the server or not in $\mathcal{C}_t$, and either using the cumulative gradient updates Eq. 12 or not. In total, we have $2 \times 2 \times 2 \times 2 = 16$ different ways of measuring gradient diversity. Therefore, we perform all the 16 different ways of calculating gradient diversity, and repeat the comparison between FedAvg and the grouping-based method under the same setting of the experiments on EMNIST. The results are reported in Fig. 6.

We can see that the gradient diversity values of the grouping-based averaging method are consistently lower than FedAvg, and the corresponding test accuracy values are consistently higher than FedAvg. More interestingly, we see that the grouping-based averaging method significantly accelerates the convergence speed compared to FedAvg. In other words, in Fig. 6 a large gradient diversity value across different users can slow down the training process significantly.

IV. Evaluating SSFL in Different Settings

In this section, we extensively evaluate our grouping-based SSFL solution, i.e., CRL objective combined with GN and grouping-based model averaging. In addition to comparing with prior work, we vary the environmental factors mentioned in § II-C, which include the non-iidness R, the communication period T, the number of labeled data N_s in the server, the user number K and the number of participating users C. All the environmental factors, as well as hyperparameters used in this section, are reported in Tab. A.2, page 19 in our full version [22] online.

Experiment settings. We consider three datasets, Cifar-10 [30], SVHN [31], and EMNIST [32] in our empirical evaluation. We use ResNet as the training model on both Cifar-10 and SVHN datasets; and we use the same CNN model as [27] on EMNIST.

A. Comparing with other supervised/semi-supervised results

In this subsection, we compare our grouping-based method with other FL algorithms, in both semi-supervised and supervised settings. First, in the semi-supervised setting, we conduct the experiment on Cifar-10 with exactly the same setting as

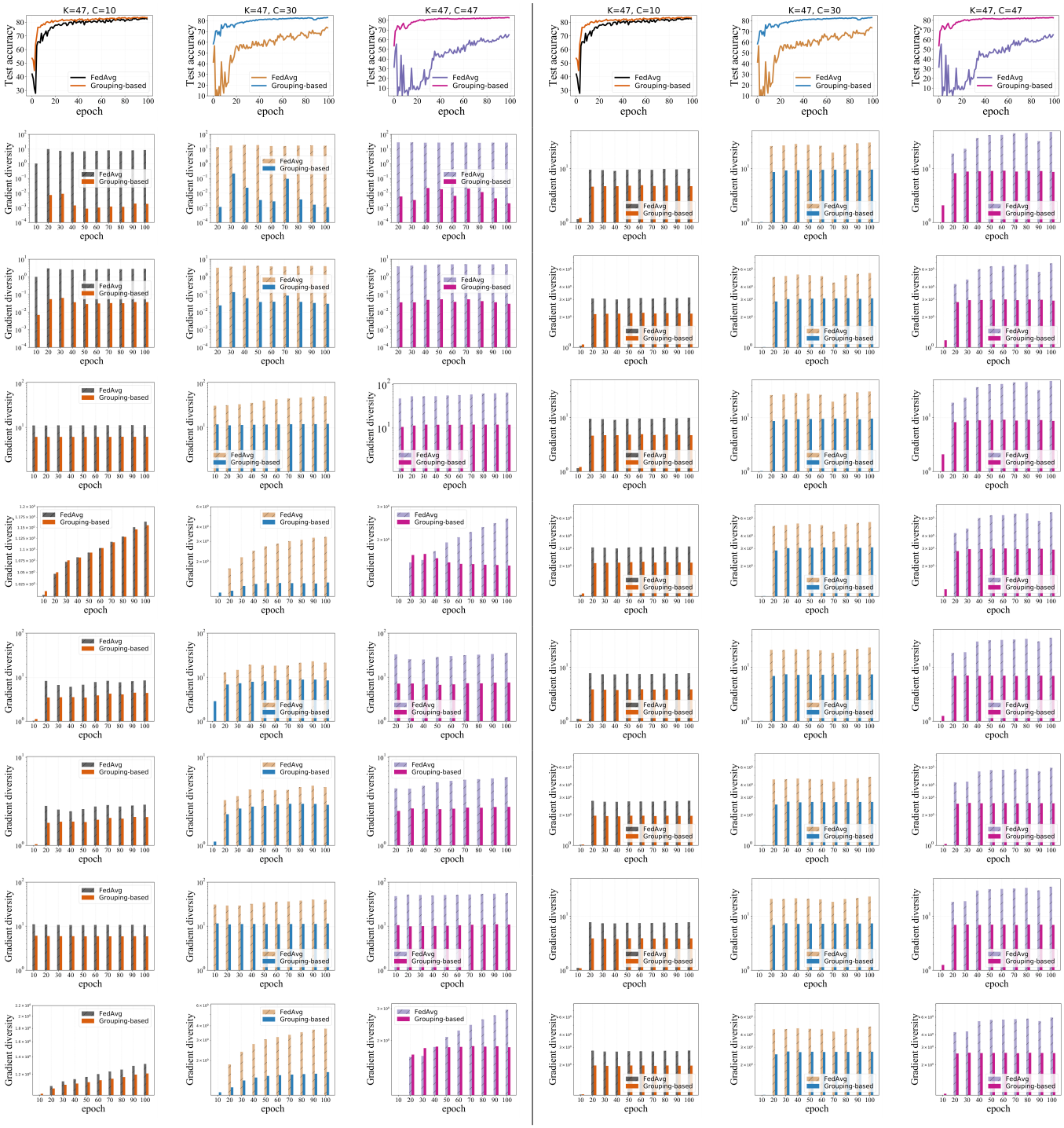

Fig. 6: **Left (1-3 column)** *(Line 1) The convergence curves on EMNIST. (Line 2) Results on gradient diversity defined in Eq. 1. (Line 3) Results on gradient diversity defined in Eq. 9. (Line 4) Results on gradient diversity defined in Eq. 1, including both the users and the server. (Line 5) Results on gradient diversity defined in Eq. 9, including both the users and the server. (Line 6) Results on gradient diversity defined in Eq. 10. (Line 7) Results on gradient diversity defined in Eq. 11. (Line 8) Results on gradient diversity defined in Eq. 10, including both the users and the server. (Line 9) Results on gradient diversity defined in Eq. 11, including both the users and the server.* **Right (4-6 column)** *(Line 1) The convergence curves on EMNIST. (Line 2) Results on gradient diversity defined in Eq. 1 with ∇w_k^t defined in Eq. 12. (Line 3) Results on gradient diversity defined in Eq. 9 with ∇w_k^t defined in Eq. 12. (Line 4) Results on gradient diversity defined in Eq. 1 with ∇w_k^t defined in Eq. 12, including both the users and the server. (Line 5) Results on gradient diversity defined in Eq. 9 with ∇w_k^t defined in Eq. 12, including both the users and the server. (Line 6) Results on gradient diversity defined in Eq. 10 with ∇w_k^t defined in Eq. 12. (Line 7) Results on gradient diversity defined in Eq. 11 with ∇w_k^t defined in Eq. 12. (Line 8) Results on gradient diversity defined in Eq. 10 with ∇w_k^t defined in Eq. 12, including both the users and the server. (Line 9) Results on gradient diversity defined in Eq. 11 with ∇w_k^t defined in Eq. 12, including both the users and the server.*

Table II: *Comparing with [9] in exactly the same setting on Cifar-10. Note that we follow [9] and use ResNet-9.*

	FedMatch	Ours
Labels-at-client (iid)	53.51%	**71.61%**
Labels-at-client (non-iid)	54.26%	**69.05%**
Labels-at-server (iid)	46.81%	**63.32%**
Labels-at-server (non-iid)	47.11%	**63.24%**

Table III: *Comparison with supervised FL. Here, "*" is calculated according to the setting in DataSharing.*

Method	Test accuracy
Supervised FedAvg	78.52% ($R = 0.29$)
DataSharing	81.82% ($R = 0.29^*$)
Grouping-based (ours)	**92.96**% ($R = 0.4$)

Table IV: *Comparison with two other supervised FL algorithms EASGD and OverlapSGD on Cifar-10.*

Method	$T = 2$	$T = 8$	$T = 32$
EASGD	91.12%	88.88%	–
OverlapSGD	91.63%	91.45%	–
Grouping-based (ours)	**94.22**%	**93.58**%	**91.92**%

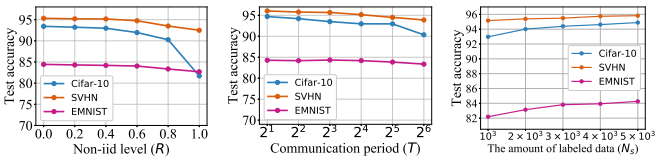

Fig. 7: *(Left) Comparison between different non-iid levels (R) on Cifar-10, SVHN and EMNIST. (Middle) Accuracy versus communication period T. (Right) Accuracy versus labeled data points in the server (N_s).*

a recent SSFL paper [9]. Note that we follow [9] and use ResNet-9. For the Cifar-10 data, according to Table 1 in [9], we set $N_s = 5000$, $K = 100$, $C = 5$, and $R = 0$ (which is the iid case) or $R = 1$ (which is the most difficult non-iid case). From Tab. II, one can see that our grouping-based solution outperforms the method proposed in [9] by a large margin. We notice that the results in [9] are presented in two different settings including the *labels-at-server* setting and the *labels-at-client* setting. The first setting is the same as our paper, i.e., only the server has labeled data, while the users have unlabeled data. In this setting, $N_s = 5000$ labeled data are own by the server. The second setting is different but it is straightforward to apply our grouping-based solution. In this setting, $N_s = 5000$ labeled data are distributed to 100 users. In each round, $C = 5$ users are random selected to communicate with the server. See Appendix H of our full version [22] for the details of adapting our solution to the label-at-the-client setting.

We also compare our solution with supervised FL methods in Tab. III. We choose two supervised FL methods for comparison: Supervised FedAvg [2] and DataSharing [16]. We set $K = 10$, $C = 10$ and $T = 32$, and we use ResNet-18 to be the model for training. The non-iid setting of DataSharing [16] corresponds to the scenario where we set $R = 0.29$. For our solutions, we set $N_s = 1000$ and $R = 0.4$. The detailed experimental parameters of different methods can be seen from rows 22-25 of Tab. A.2 in [22]. Larger R means a higher non-iid level and thus a more difficult scenario (which we have experimentally demonstrated in Fig. 7). From Tab. III we see that the performance of our method ($R = 0.4$) on Cifar-10 is still better than Supervised FedAvg ($R = 0.29$) and DataSharing methods ($R = 0.29$) even when the scenario of $R = 0.4$ is more difficult.

We also compare our method with EASGD [14] and OverlapSGD [15] which are communication efficient algorithms under supervised settings. We use the same parameters in their papers, i.e., $K = 16$, $R = 0.4$, $C = 16$ and $N_s = 1000$ on Cifar-10. See rows 26-29 of Tab. A.2 for the details. The results are shown in Tab. IV. We see that our result has better accuracy than both EASGD and OverlapSGD. Particularly, even with $T = 32$ (larger communication period T means a harder scenario), our method has 0.80% or 0.29% better performance, as compared to EASGD or OverlapSGD in the setting of $T = 2$, respectively. Note that both EASGD and OverlapSGD are supervised algorithms, which means they have all the data labels.

B. Impact of environmental factors R, T, and N_s

In this subsection, we study the effect of the first three environmental factors. First, we illustrate the effect of the non-iid level R (defined in Definition 2). For Cifar-10, SVHN and EMNIST, the experiment parameters are reported respectively in rows 1-3 of Tab. A.2 in our full version [22]. For each experiment, we fix all the parameters only except the non-iidness parameter R. The results are shown in the left of Fig. 7. When $R = 0$, each user has the same empirical class distribution which is uniform. When $R = 1$, each user only has a single class of data. As can be seen, the accuracy decreases as the non-iid level R increases (from 93.42% to 81.7% on Cifar-10, from 95.32% to 92.49% on SVHN, and from 84.43% to 82.69% on EMNIST.) This is in accord with our intuition that iid data distribution typically leads to the best result.

We also illustrate the effect of the communication period T on Cifar-10, SVHN, and EMNIST. For these three datasets, the experiment parameters are reported in rows 4-6 of Tab. A.2 in [22]. In these experiments, we again only vary the communication period T while holding all the remaining parameters fixed. The middle of Fig. 7 presents our results. Increasing T (i.e., communicating less frequently) leads to a worse generalization performance. This is explainable since the local model can overfit when T is large. In addition, the convergence curves on Cifar-10 for different T can be found in Figure B.1 of our online version [22].

Then, we investigate the impact of the number of labeled samples N_s in the server. For the experiments on three datasets, the experiment parameters are shown in rows 7-9 of Tab. A.2 in [22]. The results are shown in the right part of Fig. 7. We notice that increasing the amount of labeled data in the server can improve the final generalization performance. For example,

Table V: *Accuracy versus amount of communicating users C on Cifar-10 and SVHN. Here, "*" means we train SVHN for $E = 120$ epochs instead of $E = 40$ epochs for normal SVHN training.*

Dataset	$K = 10$, $C = 10$	$K = 20$, $C = 20$	$K = 30$, $C = 30$
Cifar-10	92.86%	92.93%	92.12%
SVHN	95.49%	94.99%	78.77% (94.93%*)
	$K = 10$, $C = 10$	$K = 20$, $C = 10$	$K = 30$, $C = 10$
Cifar-10	92.86%	93.19%	92.84%
SVHN	95.49%	95.43%	93.56%

with 5000 labeled samples, the test accuracy values on all the three datasets are higher as compared to 1000 labeled data, e.g., for Cifar-10 the improvement is 1.92%, for SVHN the improvement is 0.66%, and for EMNIST the improvement is 2.07%. These results are reasonable since the increase in the amount of labeled data can make the model trained by the server more accurate, which helps the users obtain more accurate pseudo-labels. In the extreme case where the server has the entire labeled training dataset, the situation degrades to a supervised learning setting.

C. Impact of environmental factors C and K

In this subsection, we analyze the remaining two environmental factors C and K. Again, we change one specific environmental factor while holding all the other factors fixed. The settings of the environmental factors for the experiments in this subsection are reported in rows 10-18 of Tab. A.2 in [22].

The results of Cifar-10 and SVHN are shown in Tab. V, and the result of EMNIST is presented in Tab. I. On the top of Tab. V, we set $C = K$ and increase K. At the bottom of Tab. V, we show the result with fixed $C = 10$ and various K (from 10 to 30).

As can be seen from the top of Tab. V, increasing the number of users K has a marginal effect ($<$1%) on the accuracy, from $K = 10$ to $K = 30$. One notable thing here is that with $K = 30$, if we train 40 epochs on SVHN, the accuracy is 78.77%, which is 16.72% lower than $K = 10$. If we increase the training epochs from 40 to 120 for $K = 30$ on SVHN, the final accuracy is 94.93%. One can refer to Fig.C.1 of [22] for the convergence curve of this experiment.

Similar to the results presented in Tab. I, when comparing the results at the bottom of Tab. V to the results on the top, the results when $C < K$ are consistently better than when $C = K$. Particularly, the $K = 30$, $C = 10$ case outperforms $C = 30$ by 0.72% on Cifar-10 and by 14.79% on SVHN, respectively.

D. Additional results for other datasets, environmental factors, and experimental settings

There are other important issues we consider. In this subsection, we discuss a wide range of other experiments that we have considered, and one can see Appendix E-J in our full version [22] for more details.

In Appendix E of [22], we study the impact of the user connection ratio $\eta = C/K$. We study this ratio because, in practical FL, the number of connected users can vary during training. Our results show that, when η increases to a large value (e.g. when η is close to 1), the diversity of models across users becomes too large, and the performance decreases.

In Appendix F of [22], we show the results of our grouping-based averaging in fully supervised FL (SFL) to see whether this particular way of averaging is more suitable for the semi-supervised setup or the supervised setup. We conduct experiments on EMNIST using SFL with three different settings with different number of users $K \in \{47, 20, 10\}$. In these three settings, we let $C = K$. The results show that the performance of the grouping-based averaging is only slightly better than that of FedAvg. Thus, the performance gain of the grouping-based averaging method for SFL is much less than that of SSFL. This mean that grouping-based averaging is more suitable for the semi-supervised setup than the supervised setup.

In Appendix G of [22], we compare our grouping-based solution, which works in a distributed setting, to FixMatch [11] which is originally proposed for the centralized semi-supervised setting. We see that the results of our grouping-based solution are comparable to FixMatch even if FixMatch uses centralized training.

In Appendix H of [22], we conduct an experiment on EMNIST to test the grouping-based solution in the "labels-at-client setting". Note that we have conducted the same experiment on Cifar-10, and the results can been seen from Tab. II. For EMNIST, we set $K = C = 47$, $T = 16$ and $N_s = 4700$, which has the same environmental factors as reported in Tab. I for the grouping-based solution. From the results in Appendix H, we see that for $R = 0.4$, the obtained accuracy is 81.88%. This result and the results shown in Tab. II indicate that our method can still apply to the alternative labels-at-client setting where users have both labeled and unlabeled data.

In Appendix I of [22], we test all of our solutions on STL-10 [33], which is a dataset created specifically for semi-supervised learning. The results show that self-training solution achieves 74.25%. The CRL with BN uses Eq. 4 and Eq. 3 and achieves 78.96%. Then, when we change BN to GN, we achieve 81.71%. When we further change FedAvg to grouping-based averaging, we achieve 82.81%. These results further support the superiority of our proposed SSFL method.

In Appendix J of [22], we study the performance of the FedAvg solution and the grouping-based averaging with a large user number, to see if this particular way of averaging is still useful when the number of users is particularly large (which is closer to the practical scenario). The results in Appendix J of [22] show that the performance of the grouping-based averaging is better than that of FedAvg even with 470 users. Besides, as the number of communicating users C increases, the performance of the FedAvg decreases, which is consistent with the experimental phenomenon observed in Fig. 5.

V. Related work

Federated learning. Federated learning (FL) [1], [2], [4], [16], [18], [34]–[38] is a decentralized computing framework that enables multiple users to learn a shared model while potentially protecting the privacy of users (although recent work [39] shows this may not be the case). Federated Averaging

(FedAvg) [2], which is the most popular FL algorithm, shows good performance when the data distribution across users is iid. However, in the non-iid case, the performance can significantly degrade. In fact, dealing with non-iid distributions is one of the most critical challenges in FL [16], [35], [40]. In [16], a data-sharing method is proposed to improve the final accuracy. However, sharing massive data among all users requires both large storage space as well as stable connections between users and the server. In [41], Haddadpour *et al.* show that a bounded gradient diversity is necessary to achieve fast convergence in periodic averaging. While [41] focuses on convergence analysis, we use existing/new methods to reduce the large gradient diversity that arises from both non-iid data and semi-supervised training. Importantly, all of these prior papers require the data stored by the local users to come with ground-truth labels (in order to perform model updates locally). The FL problem in the semi-supervised setting, when users do not have labels, however, is "relatively ignored" and has "little prior arts," as mentioned in a recent survey paper [4].

In addition to the challenge of the non-iidness of the data distribution and the need for local ground truth labels, communication efficiency is another critical problem in FL [37], [42]–[45]. One way to relieve the communication burden of FL is to increase the period (the number of local gradient descent iterations) between consecutive communication stages. However, when this communication period increases, the dissimilarity between different models increases, and the fusion of these models by the server may lead to accuracy degradation. To handle this problem, [43] proposes FedProx, which adds a proximal term in the user local loss function to restrict the update distance between the local model and the global model. Other work considers gradient compression and model compression to reduce the communication cost [37], [38], [45]. For example, [37] proposes atomic sparsification of stochastic gradients, which leads to significantly faster distributed training.

Semi-supervised Learning. Semi-supervised learning (SSL) is a classical problem when only a small fraction of data is labeled [11], [23], [26], [46]–[50]. SSL includes many impactful algorithms. For example, self-training [51] uses the model's own predictions on unlabeled data to supervise the training of the same model. Co-training [52] trains two models in parallel using two set of conditionally independent features, and let the two models supervise each other. Tritraining [47] first trains three classifiers using bootstrap. Then, each classifier is trained on samples agreed by the other two classifiers. Graph-based SSL [53] propagates labels on a graph generated by the similarity between different samples.

In recent years, the problem of SSL in the context of deep neural networks has been extensively studied. In [23], a specific consistency regularization is used: the average predictions on several augmented views of a single unlabeled sample is sharpened (using temperature scaling) and used to supervise the different predictions. Mixup [54] is further applied as a traditional regularization approach. Unsupervised data augmentation (UDA) [55] applies AutoAugment [56] to generate data-dependent augmentations to improve the performance. In [26], a self-training method is introduced, which improves the state-of-the-art accuracy on ImageNet [57], even compared to supervised learning [29], [58]. In [11], a simplified SSL loss is proposed which directly uses pseudo-labeling to provide consistency regularization on samples.

SSFL. Regarding the motivation of SSFL, a recent survey paper [4] raises the practical concern that users may not have ground-truth labels. Regarding the problem formulation, [9] is the most relevant. It uses a consistency loss to achieve the agreement between users, which aligns with the intuition in our method to reduce gradient diversity. The setting of [34] is also similar to ours but focuses on the label-at-client scenario. Apart from these two, there are several other contemporary papers that consider different settings. For example, the paper [59] considers using shared unlabeled data for distillation-based message exchanging. The paper [18] assumes that the unlabeled data is held by the server. The paper [60] focuses on the "vertical" FL setting in which the data is partitioned from the feature dimension. Two other papers [61], [62] use SSFL in specific professional fields. Another paper [63] studies semi-supervised private aggregation of an ensemble of teacher models trained on separate subsets of the whole dataset, which is not in the FL setting but is closely related.

VI. Conclusions

We studied the semi-supervised federated learning (SSFL) setting in which most samples are unlabeled. Based on the observations of large gradient diversity, we proposed to use GN and a novel grouping-based model averaging method. We conducted extensive evaluations in various scenarios to evaluate our solution. The results showed that our SSFL method achieves better test accuracy even when compared to existing semi-supervised or supervised FL algorithms.

It is worth noting that metrics based on the similarity between weights, feature representations, and gradients have been widely explored to analyze the trainability and the generalization performance of learning models [10], [64], [65]. For non-iid distributions in FL and the particular semi-supervised setting considered in our paper, gradient diversity is "intuitive" and is easy to measure. Our paper focuses on large-scale empirical analysis, and we aim to find algorithms that are easy to implement on today's deep neural networks. However, methods that explicitly use similarity-based metrics (e.g., as a regularization term in the loss function) might also give convincing results.

We emphasize that our solution can be extended to other FL scenarios, such as standard supervised FL (see Appendix F of our full version [22]) and the label-at-client FL [4], [9] (see Tab. II). Another challenging scenario worth mentioning is where there is a significant mismatch between the user data distributions and the distribution at the server, in which case the label supervision from the server may conflict with the information provided by users. We envision that techniques from unsupervised domain adaptation [66] are useful to address this problem. In addition, personalization [67], [68] is important for SSFL because it can mitigate the mismatch between the

data distributions at the server and at the users' side. Although our work focuses on empirical analysis, it is meaningful future work to explore the theory behind the new SSFL setting, e.g., by advancing recent theoretical results in non-iid FL [16], [69] and combining with analysis of particular data augmentation schemes such as CRL.

ACKNOWLEDGMENTS

We would like to thank Jianyu Wang and Daniel Rothchild for their valuable feedback. We would like to acknowledge DARPA, NSF, and ONR for providing partial support of this work. Michael W. Mahoney would like to acknowledge the UC Berkeley CLTC, ARO, IARPA (contract W911NF20C0035), NSF, and ONR for providing partial support of this work. Kannan Ramchandran would like to acknowledge support from NSF CIF-2007669, CIF-1703678, and CIF-2002821. Joseph E. Gonzalez would like to acknowledge supports from NSF CISE Expeditions Award CCF-1730628 and gifts from Alibaba, Amazon Web Services, Ant Group, Ericsson, Facebook, Futurewei, Google, Intel, Microsoft, Nvidia, Scotiabank, Splunk and VMware. Our conclusions do not necessarily reflect the position or the policy of our sponsors, and no official endorsement should be inferred.

REFERENCES

[1] J. Konečný, H. B. McMahan, F. X. Yu, A. T. Suresh, D. Bacon, and P. Richtárik, "Federated learning: Strategies for improving communication efficiency," 2018.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[3] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *Neural Information Processing Systems Workshop on Optimization for Machine Learning*, 2015.

[4] Y. Jin, X. Wei, Y. Liu, and Q. Yang, "Towards Utilizing Unlabeled Data in Federated Learning: A Survey and Prospective," *arXiv preprint arXiv:2002.11545*, 2020.

[5] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[6] R. Snow, B. O'connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263, 2008.

[7] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, pp. 1–210, 2021.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[9] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning," *International Conference on Learning Representations*, 2021.

[10] D. Yin, A. Pananjady, M. Lam, D. Papailiopoulos, K. Ramchandran, and P. Bartlett, "Gradient diversity: a key ingredient for scalable distributed learning," *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1998–2007, 2018.

[11] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, 2015.

[13] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018.

[14] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging sgd," in *Advances in neural information processing systems*, pp. 685–693, 2015.

[15] J. Wang, H. Liang, and G. Joshi, "Overlap local-SGD: An algorithmic approach to hide communication delays in distributed SGD," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8871–8875, 2020.

[16] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

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

[18] N. Guha, A. Talwlkar, and V. Smith, "One-shot federated learning," *NeurIPs Workshop on Machine Learning on the Phone and other Consumer Devices*, 2019.

[19] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *Conference on Neural Information Processing Systems, Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018.

[20] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[21] M. Basseville, "Distance measures for signal processing and pattern recognition," *Signal processing*, vol. 18, no. 4, pp. 349–369, 1989.

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

[23] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

[24] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[25] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, June 2020.

[26] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2020.

[27] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *International Conference on Learning Representations*, 2021.

[28] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-iid data quagmire of decentralized machine learning," *Proceedings of the International Conference on Machine Learning*, vol. 119, pp. 4387–4398, 2020.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[30] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

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

[32] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks*, pp. 2921–2926, 2017.

[33] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

[34] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," in

International Wireless Communications and Mobile Computing, pp. 1666–1671, 2020.

[35] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," *International Conference on Learning Representations*, 2020.

[36] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *International Conference on Learning Representations*, 2019.

[37] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, pp. 9850–9861, 2018.

[38] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary compression for communication-efficient federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

[39] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2019.

[40] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *Workshops on Neural Information Processing Systems*, 2018.

[41] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv preprint arXiv:1910.14425*, 2019.

[42] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

[43] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization for heterogeneous networks," *Workshop on the International Conference on Machine Learning*, 2019.

[44] X. Yao, C. Huang, and L. Sun, "Two-stream federated learning: Reduce the communication costs," in *2018 IEEE Visual Communications and Image Processing*, pp. 1–4, 2018.

[45] H. B. McMahan, D. M. Bacon, J. Konecny, and X. Yu, "Communication efficient federated learning," Nov. 7 2019. US Patent App. 16/335,695.

[46] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[47] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[48] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in neural information processing systems*, pp. 3546–3554, 2015.

[49] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, pp. 1195–1204, 2017.

[50] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *International Conference on Learning Representations*, 2020.

[51] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

[52] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998.

[53] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning*, pp. 912–919, 2003.

[54] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *International Conference on Learning Representations*, 2018.

[55] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6256–6268, 2020.

[56] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

[58] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *Proceedings of the International Conference on Machine Learning*, 2019.

[59] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data," *IEEE Transactions on Mobile Computing*, no. 01, pp. 1–1, 2021.

[60] Y. Kang, Y. Liu, and T. Chen, "Fedmvt: Semi-supervised vertical federated learning with multiview training," *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction*, 2020.

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

[62] D. Yang, Z. Xu, W. Li, A. Myronenko, H. R. Roth, S. Harmon, S. Xu, B. Turkbey, E. Turkbey, X. Wang, W. Zhu, G. Carrafiello, F. Patella, M. Cariati, H. Obinata, H. Mori, K. Tamura, P. An, B. J. Wood, and D. Xu, "Federated semi-supervised learning for covid region segmentation in chest ct using multi-national data from china, italy, japan," *Medical Image Analysis*, vol. 70, p. 101992, 2021.

[63] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *International Conference on Learning Representations*, 2017.

[64] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, pp. 3519–3529, PMLR, 2019.

[65] J. Liu, G. Jiang, Y. Bai, T. Chen, and H. Wang, "Understanding why neural networks generalize well through gsnr of parameters," *International Conference on Learning Representations*, 2020.

[66] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in neural information processing systems*, pp. 136–144, 2016.

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

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

[69] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *International Conference on Learning Representations*, 2020.