Federated Learning with Local Openset Noisy Labels

Zonglin Di[1], Zhaowei Zhu[2], Xiaoxiao Li[3,4], and Yang Liu[1]

[1] University of California, Santa Cruz, Santa Cruz, CA, 95064, USA
{zdi, yangliu}@ucsc.edu
[2] Docta.ai, San Jose, CA, 95112, USA
zzw@docta.ai
[3] University of British Columbia, Vancouver, BC, V6T 1Z4, Canada
xiaoxiao.li@ece.ubc.ca
[4] Vector Institute, Toronto, ON M5G 1M1, Canada

Abstract. Federated learning (FL) is a learning paradigm that allows the central server to learn from different data sources while keeping the data private locally. Without controlling and monitoring the local data collection process, the locally available training labels are likely noisy, *i.e.*, the collected training labels differ from the unobservable ground truth. Additionally, in heterogenous FL, each local client may only have access to a subset of label space (referred to as openset label learning), meanwhile without overlapping with others. In this work, we study the challenge of FL with local openset noisy labels. We observe that many existing solutions in the noisy label literature, *e.g.*, loss correction, are ineffective during local training due to overfitting to noisy labels and being not generalizable to openset labels. For the methods in FL, different estimated metrics are shared. To address the problems, we design a label communication mechanism that shares "contrastive labels" randomly selected from clients with the server. The privacy of the shared contrastive labels is protected by label differential privacy (DP). Both the DP guarantee and the effectiveness of our approach are theoretically guaranteed. Compared with several baseline methods, our solution shows its efficiency in several public benchmarks and real-world datasets under different noise ratios and noise models. Our code is publicly available at `https://github.com/UCSC-REAL/FedDPCont`.

Keywords: Federated Learning · Openset Noisy Labels· Weakly Supervised Learning

1 Introduction

Data heterogeneity is a common issue among different data centers. The label spaces of the data centers are likely different due to the heterogeneity of data sources. For example, the virus variants during the pandemic may differ in different regions, leading to an extremely heterogeneous data distribution among data centers. The heterogeneity challenges collaborations among data centers, *e.g.*, federated learning (FL), where each data center joins as a client to train a

uniform and stronger global model for all the regions without sharing the sensitive data. In addition to a heterogeneous label space, what makes matters worse is that the **observed label space** may be noisy due to the limited knowledge access between different data centers, making this problem more challenging. This paper aims to provide solutions for a practical FL setting where not only do each client's training labels carry different noise ratios[5], but the observed label space at these clients can also be noisy and differ, even though their underlying clean labels are drawn from the same label space. We call that such an FL system has *local openset noise* problems if the observed label space is noisy and differs across clients. Note the word "local" makes it different from the traditional definition of openset [12, 44]. From the perspective of every single client, the local label space is smaller than the global one. Therefore the local clients are dealing with an openset problem. Under the FL protocol, the model architecture is the same and shared between all the clients and the server.

The above local openset label noise will pose significant challenges if we apply the existing learning with noisy label solutions locally at each client. For instance, a good number of these existing solutions operate with centralized training data and rely on the design of robust loss functions [10, 14, 35, 37, 47, 60, 63]. Implementing these approaches often requires assumptions, which are likely to be violated if we directly employ these centralized solutions in a federated learning setting. For example, loss correction is a popular and fundamental design of robust loss functions [19, 28, 35, 37, 39, 49], where the key step is to estimate the label noise transition matrix correctly [5, 59, 64, 67]. Correctly estimating the label noise transition matrix requires observing the full label space, when the ground-truth labels are unavailable. In FL, where the transition matrix is often estimated only with the local openset noisy labels, existing estimators of the noise transition matrix would fail. Moreover, even though we can have the best estimate of the noise transition matrix if we have the ground-truth labels for the local instances, the missing of some label classes would make the estimate different from the ground-truth one, and again leads to failures (detailed example in Section 3.4).

Because of the limitation of local transition matrix estimation, the existing FL methods [50, 54] first estimate the transition matrix and share the matrices to the server and the server will apply different strategies to different servers. Similarly, we may share some label information among the clients to generalize some centralized training methods to FL. However, it is against privacy protection, making it challenging in real usage. Moreover, it is also important to figure out what kind of label information is sufficient to solve the local openset noisy problems in FL. In this paper, we use the global label distribution as a hint to local clients, where the hint is used in a contrastive way to avoid overfitting to noisy labels. To protect privacy during label communication, we randomly flip the shared labels to ensure label differential privacy (DP). Our contributions are summarized as follows.

[5] Noise ratio is the ratio of the corrupted (wrong) labels in the local dataset.

- We formally define the openset noise problem in FL, which is more practical than the existing heterogeneous noisy label assumptions. The challenges along with the openset noise are also motivated by analyzing the failure cases of the existing popular noisy learning solutions such as loss correction or the methods based on loss correction [28, 35, 37, 63].
- We propose a novel framework, FedDPCont, to solve the openset label noise problem, which builds on the idea of using globally shared private contrastive labels to avoid overfitting to local noisy labels.
- To mitigate the gap between the centralized usage of noisy labels and the federated one, we propose a *label communication* algorithm **with a differential privacy (DP) guarantee**. We also prove that benefiting from label communication, the gradient update of aggregating local loss with private labels is guaranteed to be the same as the corresponding centralized loss, and further establish its robustness to label noise.
- We empirically compare FedDPCont with several baseline methods on both benchmark datasets and practical scenarios, showing that, in terms of FL with openset label noise, directly applying centralized solutions locally cannot work and FedDPCont significantly improves the performance.

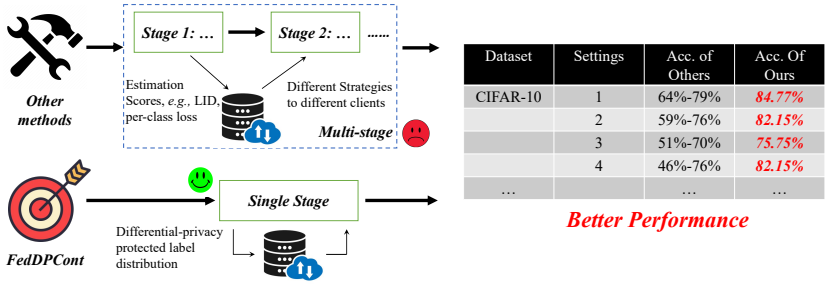


Fig. 1: FedDPCont is a simple but effective method in solving the locally openset noisy label problem with only one stage **(more simple)** but **higher** performance.

2 Related Works

2.1 Federated Learning

Federated learning is a collaborative training method to make full use of data from every client without sharing the data. FedSGD [40] is the way of FL to pass the gradient between the server and the clients. To improve the performance, FedAvg [32] is proposed and the model weight is passed between the server and the clients. In practice, openset problem is common in FL because the source of every client may vary a lot and it is very likely to find that some of the classes are unique in the specific clients. There are a lot of works to analyze and solve the non-IID problem in FL [3, 20, 24–26, 58, 62].

Table 1: Comparison with other existing FL methods for noisy label. **Non IID** stands for whether the method can be used when the data distribution among the clients is non-IID. **Single-stage** means that whether the training method remain the same in the whole training process. For example, if the method needs to estimate the noise level, the first stage will be pre-training the model and the next stage is to apply the proposed strategy, which is not single stage. **Free of Noise Estimation** refers to whether the method depends on a step of noise estimation. **No Private Set** means whether the method retains a clean private set for performance estimation or not. Our approach, FedDPCont, presents a single-stage Federated Learning (FL) method designed to address the locally openset noisy label problem without the need for noise estimation or a private set. The information shared is the label distribution that is theoretically guaranteed to be protected by differential privacy.

Method	Non IID	Single-stage	Free of Noise Est.	No Private Set	Shared Info.
FedCorr [54]	✓	✗	✗	✓	LID
FedRN [21]	✓	✓	✗	✓	Reliability score
FedNoRo [50]	✓	✗	✗	✓	Per-Class Loss
RHFL [9]	✓	✗	✗	✗	-
RFLNL [55]	✗	✓	✓	✓	Global Pseudo Labels
FedDPCont (Ours)	✓	✓	✓	✓	DP-protected Label Dist.

2.2 Label Noise Learning

Label noise is common in the real world [2, 31, 42, 48, 53, 57, 65]. Traditional works on noisy labels usually assume the label noise is *class-dependent*, where the noise transition probability from a clean class to a noisy class *only* depends on the label class. There are many statistically guaranteed solutions based on this assumption [28,30,34,35]. However, this assumption fails to model the situation where different group of data has different noise patterns [45]. For example, different clients are likely to have different noisy label spaces, resulting totally different underlying noise transitions. Existing works on FL with noisy labels mainly assume the noisy label spaces are identical across different clients [54,55]. There are other notable centralized solutions relying on the memorization effect of a large model (e.g., deep neural network) [7, 23, 27, 29, 41, 51]. However, in a FL system, simply relying on the memorization effect would fail, i.e., the model can perfectly memorize all local noisy samples during local training, since the local data is likely to be imbalanced and with a limited amount [15, 29]. The representative existing FL methods [9, 18, 21, 50, 54] to solve the openset noisy label problem are summarized in Table 1 and Section 3.5. Without the trouble of estimating the noise level globally or applying different strategies to different clients with different noise levels, the idea of contrastive labels is to punish the overfitting, which is supposed to avoid memorizing openset local noisy samples. Besides, the concept "openset" is also used in [43], where the focus is on the out-of-distribution features and their labels are called openset noise. It is different from ours since they did not focus on in-distribution mislabeled data.

3 Formulations

3.1 Federated learning

Consider a K class classification problem in a FL system with C clients. Each client $c \in [C] := \{1, \cdots, C\}$ holds a local dataset $D_c := \{(x_n^c, y_n^c)\}_{n\in[N_c]}$, where N_c is the number of instances in D_c and $N_c := \{1, \cdots, N_c\}$. Assume there is no overlap among $D_c, \forall c$. Denote the union of all the local datasets by $D := \{(x_n, y_n)\}_{n\in[N]}$. Clearly, we have $D = \cup_{c\in[C]} D_c$ and $N = \sum_{c\in[C]} N_c$. Denote by $\mathcal{D}_c$ the local data distribution, $(X^c, Y^c) \sim \mathcal{D}_c$ the local random variables of feature and label, $\mathcal{D}$ the global/centralized data distribution, and $(X, Y) \sim \mathcal{D}$ the corresponding global random variables. Denote by $\mathcal{X}$, $\mathcal{X}_c$, $\mathcal{Y}$, and $\mathcal{Y}_c$ the space of X, X_c, Y, and Y_c, respectively. FL builds on the following distributed optimization problem:

$$\arg\min_{\theta} \quad \sum_{c\in[C]} \frac{N_c}{N} \cdot L_c(\theta),$$

where $\boldsymbol{f}$ is the classifier, θ is the parameter of $\boldsymbol{f}$. $\boldsymbol{f}$ and f stand for the same model but different output. $f := \arg\max_{i\in[K]} \boldsymbol{f}$. To this end, the local training and global model average are executed iteratively. In local training, each client learns a model $\boldsymbol{f}_c : \mathcal{X} \to \mathcal{Y}$ with its local dataset D_c by minimizing the empirical loss $L_c(\theta_c)$ defined as: $L_c(\theta_c) := \frac{1}{N_c} \sum_{n\in[N_c]} \ell(\boldsymbol{f}_c(x_n^c; \theta_c), y_n^c)$, where for classification problems, the loss function is usually the cross-entropy (CE) loss: $\ell(\boldsymbol{f}(X; \theta), Y) = -\ln(f_{(X;\theta)}[Y]), Y \in [K]$, indicating taking the negative logarithm of the Y-th element of $\boldsymbol{f}$ given input X and model parameter θ. In the following global model average, each client c sends its model parameter θ_c to the central server, which is further aggregated following FedAvg [32]: $\theta = \sum_{c\in[C]} \frac{N_c}{N} \cdot \theta_c$.

3.2 Openset Noise in Federated Learning

When the label y is corrupted, the clean dataset D becomes the noisy dataset $\tilde{D} := \{(x_n, \tilde{y}_n\}_{n\in[N]}$ where $\tilde{y}_n$ is the noisy label and possibly different from y_n. The noisy data $(x_n, \tilde{y}_n)$ can be viewed as the specific point of the random variables $(X, \tilde{Y})$ which is from the distribution $\tilde{\mathcal{D}}$. Noise transition matrix T characterizes the relationship between (X, Y) and $(X, \tilde{Y})$. The shape of T is $K \times K$ where K is the number of classes in $\mathcal{D}$. The (i, j)-th element of T represents the probability of flipping a clean label $Y = i$ to noisy label $\tilde{Y} = j$, i.e., $T_{ij} := \mathbb{P}(\tilde{Y} = j | Y = i)$. If $\tilde{Y} = Y$ always holds, T is an identity matrix. Note the above definition builds on the assumption that T is class-dependent, which is a common assumption in centralized learning with noisy labels [28, 34, 35]. However, in FL, T is likely to be different for different clients (a.k.a. group-dependent [45]). Specifically, we use T to denote the *global* noise transition matrix for $\widetilde{D}$ and T_c to denote the *local* noise transition matrix for $\widetilde{D}_c$. In a practical federated learning scenario where the data across different clients are non-IID, different clients may have different label spaces. When the labels are noisy, we naturally have the following definition of openset label noise in FL.

Definition 1 (Openset noisy labels in FL)**.** *The label noise in client c is called openset if $\tilde{\mathcal{Y}}_c \neq \tilde{\mathcal{Y}}$.*

3.3 Generation of Openset noise

We propose the following noise generation process to model openset label noise in practical FL systems. Denote by $\mathbb{1}_{c,k}$ the indicator random variable that label class k is included in client c, where $\mathbb{1}_{c,k} = 1$ (w.p. $Q_{c,k}$) indicates client c has data belonging to class k and $\mathbb{1}_{c,k} = 0$, otherwise. The indicators $\{\mathbb{1}_{c,k} \mid \forall c \in [C], k \in [K]\}$ are generated independently with the probability matrix Q, where the (c,k)-th element is $Q_{c,k} := \mathbb{E}[\mathbb{1}_{c,k}]$. In practice, if all the elements in $\{\mathbb{1}_{c,k}|k \in [K]\}$ are identical, meaning the client c can observe nothing or all the classes, then $\{\mathbb{1}_{c,k}|k \in [K]\}$ will be re-generated until client c is an openset client. For example, that all the elements in $\{\mathbb{1}_{c,k}|k \in [K]\}$ are 0 indicates none of the class in the label space and vice versa.

Denote by $I_k := \{c|\mathbb{1}_{c,k} = 1, c \in [C]\}$ the set of clients that include class k. Denote by $\tilde{D}^{(k)} = \{n|\tilde{y}_n = k\}$ the indices of instances that are labeled as class k. For each $k \in [K]$, instances in $\tilde{D}^{(k)}$ will be distributed to clients with $\mathbb{1}_{c,k} = 1$ either uniformly or non-uniformly as follows.

- *Uniform allocation:* Randomly sample (without replacement) $|\tilde{D}^{(k)}|/|I_k|$ indices from $\tilde{D}^{(k)}$ and allocate the corresponding instances to client c. Repeat for all $c \in I_k$.
- *Non-uniform allocation:* Generate probabilities $\{u_c|c \in I_k\}$ from Dirichlet distribution $\mathsf{Dir}(\mathbf{1})$ with parameter $\mathbf{1} := [1, \cdots, 1]$ ($|I_k|$ values). Randomly sample (without replacement) $|\tilde{D}^{(k)}| \cdot u_c$ indices from $\tilde{D}^{(k)}$ and allocate the corresponding instances to client c. Repeat for all $c \in I_k$.

In this way, all the clients have openset label noise, i.e., $\mathcal{Y}_c \neq \tilde{\mathcal{Y}}, \forall c \in [C]$.

3.4 Why is Locally Openset Noisy Label Challenging using Local Transition Matrix Estimation?

Transition matrix estimation is a fundamental tool to build the noisy label methods [64, 66, 67]. Consider the following example. For a data distribution $(X, Y) \sim \mathcal{D}$ where $Y \in \mathcal{Y} := \{1, 2, \cdots, K\}$, the set of all the opensets is the combination of $\mathcal{Y}$ except the full set of $\mathcal{Y}$ and the empty set. For example, if $\mathcal{Y}$ is $\{1, 2, 3\}$, there would be $2^K - 2 = 6$ different combinations of the noisy label space: $\{1, 2, 3, (1, 2), (1, 3), (2, 3)\}$. It should be noted that it is still possible that the union of all the clients still cannot cover $\mathcal{Y}$. An example of the real and openset T in the 3-class classification problem is as follows. Suppose the *real* noise transition matrix T_{real} is shown on the LHS. However, if we only observe $\tilde{\mathcal{Y}}_c = \{1, 2\}$ in client c, the *optimal estimate* of T relying only on $\tilde{D}_c$ could only be T_{OptEst} even though we know D_c. This is because when $\tilde{\mathcal{Y}}_c = \{1, 2\}$, we have $\mathbb{P}(\tilde{Y} = 3) = 0 \Rightarrow \mathbb{P}(\tilde{Y} = 3|Y = 3) = 0$, resulting that the other two probabilities have to be normalized from $(1/16, 3/16)$ to $(1/4, 3/4)$ to get a total probability of 1.

$$T_{\text{real}} = \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 2/3 & 0 \\ 1/16 & 3/16 & 3/4 \end{bmatrix}, \; T_{\text{OptEst}} = \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 2/3 & 0 \\ 1/4 & 3/4 & 0 \end{bmatrix}$$

A good number of correction approaches in the learning with noisy labels literature would require using the transition matrix T. For instance, loss correction [37] is a popular tool to solve the noisy label problem as

$$\ell^{\rightarrow}(\boldsymbol{f}(X), \tilde{Y}) := \ell(T^{\top}\boldsymbol{f}(X), \tilde{Y}) \tag{1}$$

where $T^{\top}$ is the transpose of T. The key step of the loss correction approach is to estimate a correct T. However, if the label space is openset, the best estimated T will lead to a wrong prediction result. Based on the example above, the best corrected output is

$$T^{\top}\boldsymbol{f}(X) = \begin{bmatrix} 1 & 1/3 & 1/4 \\ 0 & 2/3 & 3/4 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(X;\theta) \\ f_2(X;\theta) \\ f_3(X;\theta) \end{bmatrix} = \begin{bmatrix} f_1(X;\theta) + f_2(X;\theta)/3 + f_3(X;\theta)/4 \\ 2f_2(X;\theta)/3 + 3f_3(X;\theta)/4 \\ 0 \end{bmatrix},$$

where $\boldsymbol{f} = [f_1, f_2, f_3]^{\top}$ and f_n is the n-th element of $\boldsymbol{f}$. The model cannot distinguish class 3 which is reasonable. However, it will misclassify class 2 to class 3 because class 3 has a larger weight. For example, given an instance $(x, y = 2)$, the cross entropy loss is $-\ln(2f_2(x;\theta)/3 + 3f_3(x;\theta)/4)$ where $f_3(x;\theta) = 1$ leads to the minimization of the loss, making the loss correction fail.

3.5 How Locally Openset Noisy Label Problem is Solved in FL?

We summarize the existing FL methods solving locally openset noisy label in Table 1. Most of the methods solve openset problems by sharing the information to the server or across the clients to get the correct observation of the limited noisy label space and get a "pre-trained" model to estimate the noise ratio, identify the noise level of the clients and apply different strategies to the clients, which is the multi-stage method. Robust FL with noisy labels (RFLNL) [55] is an FL noisy label method only for IID dataset, which natrually cannot solve the openset noisy label problem in FL. Our method, FedDPCont, is a single-stage FL method for openset (extremely non-IID) noisy labels without noise estimation or a private set. The information shared by FedDPCont is the provable differential privacy protected label distribution.

4 Proposed Method

4.1 Motivation

The main difficulty of local openset label noise exists in the mismatch of clean and noisy label spaces within a local client. Changing the label space is challenging in FL since it often requires sharing data with the server or among clients.

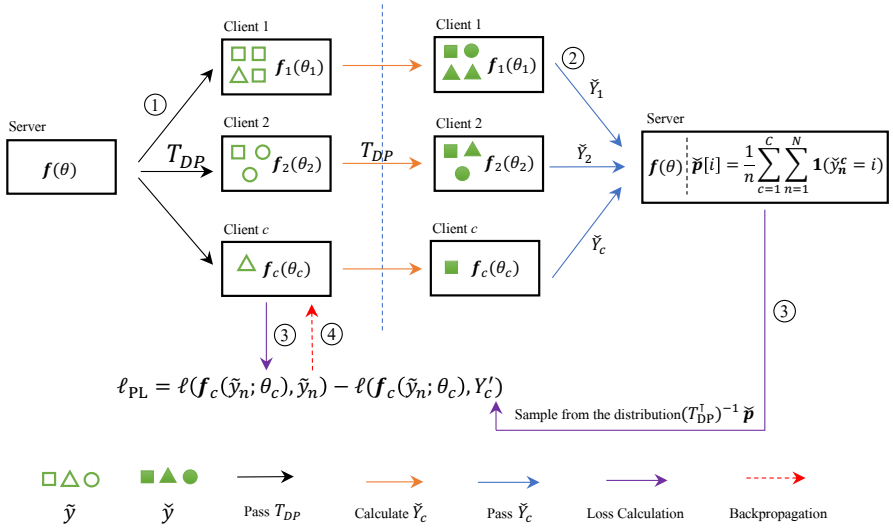


Fig. 2: The framework of FedDPCont. **Step 1** is the T_{DP} generation where the server generates T_{DP} according to ϵ and sends it to each client. After receiving T_{DP}, **Step 2** is the label communication. Every client c calculates DP label $\check{Y}_c$ according to T_{DP} and the noisy label $\tilde{Y}_c$. Clients send $\check{Y}_c$ to the server. The server aggregates every $\check{Y}_c$, calculates the posterior label distribution $\check{\boldsymbol{p}}$ and sends $(T_{\mathrm{DP}}^{\top})^{-1}\check{\boldsymbol{p}}$ to every client for the contrastive term sampling. **Step 3** is the loss calculation using the noisy label $\tilde{Y}_c$ on every client c, the model prediction $\hat{Y}_c$ and Y_c' sampled from $(T_{\mathrm{DP}}^{\top})^{-1}\check{\boldsymbol{p}}$ and calculate loss. **Step 4** is the back-propagation for contrastive gradient updates.

Therefore, we need to solve two technical challenges here: 1) What kind of information can be shared to mitigate the heterogeneity introduced by local openset label noise? 2) How do we use the shared information to help training?

For the first challenge, we consider sharing the "private labels" since only sharing the label without disclosing features is usually less sensitive than sharing features in many cases, *e.g.*, face recognition. Additionally, it is relatively easier to protect the label privacy by random responses [13]. For the second challenge, given only the private labels, we propose to use them "contrastively" to punish the overfitting of noisy labels. Intuitively, for a multi-class classification task, *e.g.*, 10 classes, a randomly picked *private label* $\check{y}_{n'}$ is likely to be a wrong label for a randomly picked feature x_n. Therefore, rather than guiding the model to memorize this pattern, we can just use it contrastively or negatively, i.e., $-\ell(\boldsymbol{f}(x_n),\check{y}_{n'})$. Therefore, the new loss function with Private Labels (PL) becomes

$$\ell_{\mathrm{PL}}(\boldsymbol{f}(x_n),\tilde{y}_n) := \ell(\boldsymbol{f}(x_n),\tilde{y}_n) - \ell(\boldsymbol{f}(x_n),\check{y}_{n'}). \tag{2}$$

The design is related to works such as [7, 30, 46], while the key difference is the selection of the labels for the second term, i.e., the private labels are drawn from the whole label space while directly using the above approach requires getting

labels locally. Intuitively, the "new" label has to be sampled globally; otherwise, the global information is missing and the negative effect of local openset label noise would induce performance degradation. Additionally, label communications in FL should be private. We defer the detailed explanation of its necessity to Appendix.

We propose the following label communication-aided algorithm FedDPCont, which we also illustrate in Fig. 2. There are *two critical components* to guarantee the success of the proposed methods with good DP protection:

- **Component 1:** Privacy-preserving global label communication given in Section 4.2.
- **Component 2:** Contrastive gradient updates at the local client using ℓ_{PL} given in Section 4.3 and the shared label information from Stage 1.

It should be noted that these both components will be executed sequentially in every communication round and the every global epoch remains the same, which illustrate the single-stage in Table 1 and Fig. 1.

4.2 Label Communication

Label privacy protection is an essential feature of FL so we cannot pass $\tilde{Y}$ to the other clients, directly. To protect privacy, we adopt the label differential privacy (DP) as Definition 2.

Definition 2 (Label Differential Privacy [13])**.** *Let $\epsilon > 0$. A randomized algorithm $\mathcal{A}$ is said to be ϵ-label differentially private (ϵ-labelDP) if for any two training datasets D and D' that differ in the label of a single example, and for any subset S of outputs of $\mathcal{A}$,*

$$\mathbb{P}(\mathcal{A}(D) \in S) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{A}(D') \in S).$$

The high-level idea is to achieve label privacy (DP), each client c will use a symmetric noise transition matrix T_{DP} to flip their local labels to protect their labelDP:

$$T_{\mathrm{DP}}[y, \tilde{y}] := \mathbb{P}(\widetilde{Y} = \tilde{y} | Y = y) = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+K-1}, \text{ if } \tilde{y} = y, \\ \frac{1}{e^{\epsilon}+K-1}, \text{ if } \tilde{y} \neq y. \end{cases}$$

where K is the number of classes. Then only the flipped labels are shared between the clients and the server. It is easy to show that sharing the flipped labels using T_{DP} suffices to preserve labelDP:

Theorem 1 (Label Privacy in FedDPCont). *Label sharing in FedDPCont is ϵ-labelDP.*

Denote by $\tilde{\boldsymbol{p}}_n^c$ the one-hot encoding of $\tilde{y}_n^c$. The whole label communication process is presented in Algorithm 1. At the beginning of the algorithm, the server will initialize T_{DP} according to ϵ and broadcast T_{DP} to all C clients. For each client c, it calculates the DP label distribution of every data point $(x_n^c, \tilde{y}_n^c)$ as $\check{\boldsymbol{p}}_n^c = T_{\mathrm{DP}}^{\top} \tilde{\boldsymbol{p}}_n^c$, where $\check{\boldsymbol{p}}_n^c$ is the distribution of DP label in client c. With this distribution, the client generates the DP private label $\check{y}_n^c, n \in [N_c]$ for every data

Algorithm 1 Label Communication in FedDPCont

Initialization: The server initialize T_{DP} according to ϵ and broadcast T_{DP} to all clients.
 # Client label differential privacy protection
for c in C clients **do**
  calculate $\check{\boldsymbol{p}}_n^c = T_{\mathrm{DP}}^\top \tilde{\boldsymbol{p}}_n^c, \forall n \in [N_c]$.
  generate the private label $\check{y}_n^c$ using $\mathbb{P}(\check{y}_n^c = i) = \check{\boldsymbol{p}}_n^c[i], \forall i \in [K], n \in [N_c]$.
  send $\{\check{y}_n^c\}_{n\in[N_c]}$ to the server
end for
The server aggregates the label $\{\check{y}_n^c\}_{n\in[N_c]}$ sent from all C clients.
The server calculates the posterior label distribution $\check{\boldsymbol{p}}$:

$$\check{\boldsymbol{p}}[i] := \frac{1}{N}\sum_{c=1}^{C}\sum_{n=1}^{N} \mathbb{1}(\check{y}_n^c = i).$$

The server calculates $(T_{\mathrm{DP}}^\top)^{-1}\check{\boldsymbol{p}}$ and sends it to each client c.
The client c samples the $\tilde{y}_{n'}$ in Eqn. (2) following $\mathbb{P}(\tilde{Y} = \tilde{y}_{n'}) = ((T_{\mathrm{DP}}^\top)^{-1}\check{\boldsymbol{p}})[\tilde{y}_{n'}]$.

point and every client sends all $\check{y}_n^c$ back to the server. After obtaining all $\check{y}_n^c$ from the clients, the server aggregates the label and calculates the posterior label distribution $\check{\boldsymbol{p}}$. To restore the correct distribution of $\tilde{Y}$, the server calculates $(T_{\mathrm{DP}}^\top)^{-1}\check{\boldsymbol{p}}$. Note that $(T_{DP}^\top)^{-1}T_{DP}^\top(\sum_{i=1}^{C}\tilde{\boldsymbol{p}}_n^c)/C = \tilde{\boldsymbol{p}}$. To apply T_{DP} and $(T_{\mathrm{DP}})^{-1}$ sequentially, FedDPCont enables the clients to share the information with the others while DP is guaranteed. Finally, the client calculates the local loss according to Equation (2) where $\tilde{Y}$ is sampled from $\mathbb{P}(\tilde{Y} = i) := ((T_{\mathrm{DP}}^\top)^{-1}\check{\boldsymbol{p}})[i]$. This label communication procedures guarantees ϵ-DP.

4.3 FedDPCont

Based on the distribution $\mathbb{P}(\tilde{Y}|\tilde{D})$, we propose *FedDPCont*, a novel framework based on FedAvg, to solve the local openset noise problem. $\mathbb{P}(\tilde{Y}|\tilde{D})$ represents the noisy label distribution given a corrupted dataset $\tilde{D}$. Denote by $\Delta_c^{(r)} := \theta_c^{r+1} - \theta_c^r$, the variation of model parameters in the r-th round of the local training in client c. Recall θ_c is the parameter of $\boldsymbol{f}_c$.

Denote by $\Delta^{(r)} := \theta^{r+1} - \theta^r$ the variation of model parameters in the r-th round of the corresponding global gradient descent update assuming the local data are collected to a central server. Define $\mathbb{P}(\mathcal{D}_c|\mathcal{D}) := \mathbb{P}((X,Y) \sim \mathcal{D}_c \mid (X,Y) \sim \mathcal{D})$. Numerically, it is calculated as N_c/N for client c given D. We have the following theorem for the calibration property of FedDPCont.

Theorem 2 (Local clients with FedAvg). *The aggregated model update of FedDPCont is the same as the corresponding centralized model update, i.e.,*

$$\sum_{c\in[C]} \mathbb{P}(\mathcal{D}_c|\mathcal{D}) \cdot \Delta_c^{(r)} = \Delta^{(r)},$$

Theorem 2 shows that the extra effect of local openset label noise can be mitigated by sharing private labels and FedAvg. Note the theorem only discusses the case in the expectation level (infinite data size), meaning the gap between distributed learning and centralized learning given limited data still exists. Given Theorem 2, we can further show ℓ_{PL} is robust to label noise as what has been done for centralized training [30].

Algorithm 2 FedDPCont.

Server: initialize model $\boldsymbol{f}_g$, global step size α_g and global communication round R.

Each Client c**:** initialize model $\boldsymbol{f}_c$, the dataset $D_c = \{(x_n^c, \tilde{y}_n^c)\}_{n\in[N_c]}$, local learning rate α_c and local updating iterations E.
The server generates and broadcasts T_{DP} to all clients according to Definition 2.
Clients generate DP labels $\check{y}_n^c$ and send $\check{y}_n^c$ to the server according to Section 4.2.
The server aggregates $\check{y}_n^c$ and calculate the posterior label distribution $\check{\boldsymbol{p}}$.
The server send $(T_{\mathrm{DP}}^{\top})^{-1}\check{\boldsymbol{p}}$ to each client.
for $i = 1 \to R$ **do**
  Randomly select C' clients from C according to the federated fraction λ
  for c in C' clients **do**
    $\boldsymbol{f}_c \leftarrow \boldsymbol{f}_g$
    for $j = 1 \to E$ **do**
      $\hat{y}_n^c \leftarrow \boldsymbol{f}_c(x_n^c), \forall n \in [N_c]$
      Sample $(y_c^n)'$ following $(T_{\mathrm{DP}}^{\top})^{-1}\check{\boldsymbol{p}}, \forall n \in [N_c]$.
      $L_c \leftarrow \frac{1}{N_c}\sum_{n=1}^{N_c}(\ell(\hat{y}_n^c, \tilde{y}_n^c) - \ell(\hat{y}_n^c, (y_c^n)'))$
      $\boldsymbol{f}_c \leftarrow \boldsymbol{f}_c - \alpha_c \cdot \nabla L_c$
    end for
  end for
  $\boldsymbol{f}_g \leftarrow \boldsymbol{f}_g - \alpha_g \cdot \sum_{c=1}^{C'}(\boldsymbol{f}_c - \boldsymbol{f}_g)$
end for

4.4 Privacy Issue

We are aware that the label distribution recovered by our algorithm may also be a concern of privacy. However, the existing works about the attack in federated learning are mainly from embedding layers [33], fully-connected layers [11,36,61], and model gradients [4, 33]. Different from the leakage of individual labels, the recovered label distribution by our algorithm has much less information. There is no direct evidence of the harm of leaking an imperfect label distribution to the best of our knowledge. In Table 4, we will illustrate that different DP privacy level ϵ corresponds to different performance, indicating that, even though we have restored the distribution of $\widetilde{Y}$ (Algorithm 1, Line 9), it is still different from the original one.

Table 2: The performance (the best accuracy) of all methods on CIFAR-10. FedDPCont is always the best method. The similar observation can be found for CIFAR-100 dataset which is given in Table 6.

Dataset	Methods	Symmetric				Random			
		0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
CIFAR-10	FedAvg	$76.84_{\pm 0.91}$	$63.34_{\pm 1.82}$	$43.83_{\pm 0.51}$	$22.13_{\pm 1.25}$	$76.24_{\pm 1.58}$	$59.19_{\pm 1.01}$	$46.80_{\pm 2.63}$	$21.80_{\pm 0.28}$
	LC	$79.14_{\pm 0.35}$	$63.57_{\pm 0.61}$	$44.33_{\pm 1.13}$	$22.98_{\pm 1.60}$	$74.96_{\pm 1.92}$	$61.49_{\pm 3.02}$	$40.52_{\pm 2.18}$	$23.84_{\pm 3.37}$
	FedProx	$70.54_{\pm 0.57}$	$59.35_{\pm 0.65}$	$45.61_{\pm 0.97}$	$22.70_{\pm 1.10}$	$68.51_{\pm 0.92}$	$58.61_{\pm 0.38}$	$43.97_{\pm 1.06}$	$24.64_{\pm 2.59}$
	Co-teaching	$78.64_{\pm 0.45}$	$70.60_{\pm 0.47}$	$48.63_{\pm 0.57}$	$21.06_{\pm 2.10}$	$75.11_{\pm 0.39}$	$59.00_{\pm 1.19}$	$31.30_{\pm 2.03}$	$17.10_{\pm 3.78}$
	T-revision	$69.16_{\pm 6.20}$	$51.86_{\pm 6.64}$	$31.93_{\pm 2.56}$	$15.27_{\pm 1.87}$	$64.69_{\pm 5.08}$	$46.22_{\pm 1.17}$	$31.81_{\pm 2.83}$	$17.12_{\pm 0.73}$
	FedDyn	$70.88_{\pm 0.77}$	$58.58_{\pm 1.14}$	$42.83_{\pm 1.23}$	$20.70_{\pm 1.66}$	$70.13_{\pm 0.99}$	$58.91_{\pm 3.06}$	$42.11_{\pm 2.84}$	$25.21_{\pm 2.98}$
	FedBN	$67.82_{\pm 0.91}$	$53.49_{\pm 0.85}$	$39.33_{\pm 2.52}$	$19.50_{\pm 0.99}$	$66.66_{\pm 4.69}$	$58.20_{\pm 1.58}$	$41.38_{\pm 1.89}$	$22.66_{\pm 2.03}$
	Scaffold	$64.02_{\pm 0.13}$	$55.50_{\pm 0.96}$	$37.48_{\pm 2.16}$	$15.10_{\pm 0.43}$	$59.13_{\pm 0.83}$	$50.36_{\pm 1.54}$	$34.73_{\pm 4.12}$	$18.23_{\pm 1.66}$
	FedCorr	$73.33_{\pm 4.82}$	$62.68_{\pm 1.18}$	$45.21_{\pm 2.25}$	$21.70_{\pm 0.30}$	$70.14_{\pm 4.01}$	$55.42_{\pm 7.26}$	$39.64_{\pm 5.43}$	$19.47_{\pm 0.97}$
	FedRN	$73.00_{\pm 0.20}$	$55.75_{\pm 0.24}$	$41.31_{\pm 1.24}$	$20.98_{\pm 0.57}$	$67.77_{\pm 1.13}$	$52.66_{\pm 1.11}$	$36.33_{\pm 0.91}$	$19.01_{\pm 0.79}$
	RoFL	$73.22_{\pm 0.04}$	$68.16_{\pm 3.19}$	$46.63_{\pm 1.85}$	$24.23_{\pm 0.45}$	$65.90_{\pm 2.82}$	$59.73_{\pm 3.60}$	$37.39_{\pm 1.34}$	$20.26_{\pm 1.56}$
	FedNoRo	$57.49_{\pm 0.88}$	$43.97_{\pm 1.73}$	$22.15_{\pm 2.89}$	$14.10_{\pm 1.50}$	$51.64_{\pm 2.13}$	$34.16_{\pm 0.61}$	$24.63_{\pm 0.44}$	$15.07_{\pm 2.40}$
	RHFL	$53.49_{\pm 0.57}$	$40.17_{\pm 0.95}$	$21.90_{\pm 1.15}$	$12.45_{\pm 0.24}$	$49.32_{\pm 1.55}$	$30.80_{\pm 0.47}$	$20.11_{\pm 0.28}$	$10.98_{\pm 0.13}$
	FedDPCont	$\mathbf{84.77}_{\pm 0.12}$	$\mathbf{75.75}_{\pm 1.96}$	$\mathbf{55.50}_{\pm 1.33}$	$\mathbf{24.64}_{\pm 0.55}$	$\mathbf{82.15}_{\pm 0.24}$	$\mathbf{72.69}_{\pm 1.57}$	$\mathbf{54.06}_{\pm 1.38}$	$\mathbf{27.55}_{\pm 1.49}$

5 Experiments and Results

5.1 Experiments Setup

To validate the generality and effectiveness of FedDPCont, we select several public datasets with various levels of difficulties, including CIFAR-10, CIFAR-100 [22] as benchmark datasets and CIFAR-N [48], Clothing-1M [53] as real-world datasets. To simulate the practical usage, we first apply the noise on the label and generate the openset candidates according to the number of classes K for every client because only the noisy label is visible to the client in the real world. On CIFAR-10 and CIFAR-100, we apply the symmetric noise for benchmark testing while we apply random noise for practical simulation. Furthermore, we also test the performance using Clothing-1M and CIFAR-N to test the performance of FedDPCont in real-world scenarios.

For baseline methods, we use FedAvg [32], forward loss correction (LC) [37], FedProx [24], Co-teaching [16] and T-revision [52], FedBN [26], FedDyn [1], Scaffold [20], FedCorr [54] and FedRN [21]. We are aware that there are other noisy learning methods that achieve impressive performance, e.g., DivideMix [23]. However, their underlying semi-supervised learning mechanisms and mix-up data augmentation [56] methods introduce massive training cost and are out of the scope of this paper. We leave discussions related to the computation cost and performance comparisons with such method to Section 12.2 in Appendix. We exclude the mixup method for the fair comparison. The local updating iteration E is 5 and the federated fraction λ is 0.1. The architecture of the network is ResNet-18 [17] for CIFAR dataset and ResNet-50 [17] with ImageNet [8] pre-trained weight for Clothing-1M. The local learning rate α_l is 0.01 and the batch size is 32. The total communication round with the server R is 300 and differential privacy ϵ are 0.81, 3.21 and 1.17 for CIFAR-10, CIFAR-100 and Clothing-1M, respectively to keep $e^{\epsilon}/(e^{\epsilon}+K-1)$ 0.2 in Section 4.2. All the experiments are

Table 3: The performance (the best accuracy) of all the methods on CIFAR-N and Clothing-1M.

Datasets	CIFAR-10			CIFAR-100	Clothing-1M
Methods	Worst	Random	Aggregate	Fine	1M Noisy Training
FedAvg	$46.55_{\pm 7.82}$	$59.69_{\pm 4.88}$	$66.41_{\pm 6.52}$	$22.65_{\pm 2.29}$	70.27
LC	$46.67_{\pm 8.21}$	$59.27_{\pm 5.72}$	$67.27_{\pm 4.76}$	$22.59_{\pm 1.66}$	70.05
FedProx	$58.47_{\pm 0.97}$	$69.35_{\pm 0.62}$	$74.48_{\pm 1.00}$	$35.33_{\pm 0.35}$	65.96
Co-teaching	$24.80_{\pm 2.27}$	$47.34_{\pm 21.05}$	$62.04_{\pm 11.26}$	$17.83_{\pm 0.39}$	40.33
T-revision	$57.85_{\pm 19.44}$	$55.06_{\pm 8.40}$	$63.40_{\pm 9.99}$	$22.18_{\pm 1.44}$	66.95
FedBN	$63.07_{\pm 3.29}$	$73.02_{\pm 1.45}$	$77.55_{\pm 2.16}$	$37.59_{\pm 0.61}$	67.43
FedCorr	$49.36_{\pm 4.66}$	$58.18_{\pm 9.97}$	$64.06_{\pm 7.58}$	$50.42_{\pm 0.37}$	69.55
FedRN	$49.96_{\pm 0.86}$	$59.70_{\pm 0.69}$	$64.08_{\pm 1.00}$	$43.31_{\pm 0.20}$	62.78
FedDPCont	$\mathbf{63.50}_{\pm 5.63}$	$\mathbf{73.68}_{\pm 4.35}$	$\mathbf{81.86}_{\pm 1.09}$	$\mathbf{52.60}_{\pm 1.91}$	**70.88**

Table 4: The influence of different ϵ on the performance on CIFAR-10 corrupted by random noise whose ratio is 0.4, where *No DP* indicates the upper bound of FedDPCont.

$\epsilon = 0.01$	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 100$	$\epsilon = 0.81$	No DP
$75.19_{\pm 1.00}$	$72.47_{\pm 2.64}$	$71.60_{\pm 1.96}$	$72.27_{\pm 1.87}$	$73.00_{\pm 1.96}$	$73.75_{\pm 2.38}$	$72.44_{\pm 1.52}$	$74.03_{\pm 1.19}$

run with 3 different random seeds to validate the generality of our methods. The details of the implementation of every baseline method in the FL setting can be found in Section 9 in the Appendix.

5.2 Synthetic Open-Set Label Noise

There are two strategies:

- **Symmetric:** We first add symmetric label noise [16,52] to dataset D and get $\tilde{D}$, then distribute $\tilde{D}$ to $\tilde{D}_c, \forall c$ following the uniform allocation in Section 3.2. The transition matrix T for the symmetric label noise satisfies $T_{ij} = \eta/(K-1), \forall i \neq j$ and $T_{ii} = 1-\eta, \forall i \in [K]$, where $\eta \in \{0.2, 0.4, 0.6, 0.8\}$ is the average noise rate.
- **Random:** We first add random label noise [67] to dataset D and get $\tilde{D}$, then distribute $\tilde{D}$ to $\tilde{D}_c, \forall c$ following the non-uniform allocation in Section 3.2. The T of random noise is generated as follows. The diagonal elements of T for the random label noise is generated by $\eta + \mathsf{Unif}(-0.05, 0.05)$, where η is the average noise rate, $\mathsf{Unif}(-0.05, 0.05)$ is the uniform distribution bounded by -0.05 and 0.05. The off-diagonal elements in each row of T follow the Dirichlet distribution $(1-T_{ii}) \cdot \mathsf{Dir}(\mathbf{1})$, where $\mathbf{1} = [1, \cdots, 1]$ ($K-1$ values). The random strategy is more practical than the symmetric one.

Results and Discussion Table 2 shows FedDPCont is significantly better than all the baseline methods in the symmetric strategy across all the noise rate

settings. It is also better than the other methods in all the settings of the random strategy. FedDPCont is very competitive in all the settings. Table 2 also shows directly applying the methods for centralized learning with noisy labels cannot be statistically better than the traditional federated learning solution (FedAvg) and its adapted version (FedProx), indicating the openset label noise in FL is indeed challenging and special treatments are necessary to generalize the centralized solution to the FL setting.

5.3 Real-World Label Noise

We also test the performance on two real-world datasets: CIFAR-N [48] and Clothing-1M [53]. Different from the benchmark datasets, these datasets are corrupted naturally. Clothing-1M is collected from the real website where both data and labels are from the real users. The noisy ratio is about 0.4 in Clothing-1M. CIFAR-N consists of CIFAR-10 and CIFAR-100. $\tilde{D}_c$ is generated according to the random setting given in Section 5.2. The labels of CIFAR-N are collected from the human annotation. There are three levels of noisy ratio in CIFAR-10, *worst*, *aggregate* and *random* while there is only one noisy level in CIFAR-100. It can be found that FedDPCont outperforms all the baseline methods in the real-world dataset, showing great potential in practical usage.

5.4 Effect of DP Level

According to Section 4.2 and 4.3, label communication and peer gradient updates at local clients are two key steps in FedDPCont. ϵ is the parameter to control the level of DP protection. Following [13], we study the influence of ϵ on the performance. We select the CIFAR-10 corrupted by random noise whose ratio is 0.4. All the experiments are run with 10 random seeds. In terms of the randomness of model initialization and the noise generation, it can be found that FedDPCont is stable with the change of ϵ, which agrees with our theoretical guarantee. If we skip the DP process and directly pass the label distribution to the server, FedDPCont reaches its upper bound. We select the corresponding upper bound of FedDPCont in Table 4. Although the upper bound will be slightly higher than that with DP process, there is no big difference regarding the performance and the security can also be guaranteed, showing the superiority of the FedDPCont design.

6 Conclusion

We have defined openset label noise in FL and proposed FedDPCont to use globally communicated contrastive labels to prevent local models from memorizing openset noise patterns. We have proved that FedDPCont is able to approximate a centralized solution with strong theoretical guarantees. Our experiments also verified the advantage of FedDPCont. Future works include testing FedDPCont with real-world FL data partitions and real-world clients such as mobile devices.

Acknowledgements

Z. Di and Y. Liu are partially supported by the National Science Foundation (NSF) under grants IIS-2007951 and IIS-2143895. X. Li is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263 (2021)
2. Agarwal, V., Podchiyska, T., Banda, J.M., Goel, V., Leung, T.I., Minty, E.P., Sweeney, T.E., Gyang, E., Shah, N.H.: Learning statistical models of phenotypes using noisy labeled training data. Journal of the American Medical Informatics Association **23**(6), 1166–1173 (2016)
3. Andreux, M., Terrail, J.O.d., Beguier, C., Tramel, E.W.: Siloed federated learning for multi-centric histopathology datasets. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning, pp. 129–139. Springer (2020)
4. Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security **13**(5), 1333–1345 (2017)
5. Bae, H., Shin, S., Na, B., Jang, J., Song, K., Moon, I.C.: From noisy prediction to true label: Noisy prediction calibration via generative model. In: International Conference on Machine Learning. pp. 1277–1297. PMLR (2022)
6. Chen, D., Gao, D., Kuang, W., Li, Y., Ding, B.: pfl-bench: A comprehensive benchmark for personalized federated learning. Advances in Neural Information Processing Systems **35**, 9344–9360 (2022)
7. Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., Liu, Y.: Learning with instance-dependent label noise: A sample sieve approach. arXiv preprint arXiv:2010.02347 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Fang, X., Ye, M.: Robust federated learning with noisy and heterogeneous clients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10072–10081 (2022)
10. Feng, L., Shu, S., Lin, Z., Lv, F., Li, L., An, B.: Can cross entropy loss be robust to label noise? In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. pp. 2206–2212 (2021)
11. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients-how easy is it to break privacy in federated learning? Advances in Neural Information Processing Systems **33**, 16937–16947 (2020)
12. Geng, C., Huang, S.j., Chen, S.: Recent advances in open set recognition: A survey. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3614–3631 (2020)
13. Ghazi, B., Golowich, N., Kumar, R., Manurangsi, P., Zhang, C.: Deep learning with label differential privacy. Advances in Neural Information Processing Systems **34**, 27131–27145 (2021)

14. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)
15. Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I.W., Kwok, J.T., Sugiyama, M.: A survey of label-noise representation learning: Past, present and future. arXiv preprint arXiv:2011.04406 (2020)
16. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems **31** (2018)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
18. Ji, X., Zhu, Z., Xi, W., Gadyatskaya, O., Song, Z., Cai, Y., Liu, Y.: Fedfixer: Mitigating heterogeneous label noise in federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 12830–12838 (2024)
19. Jiang, Z., Zhou, K., Liu, Z., Li, L., Chen, R., Choi, S.H., Hu, X.: An information fusion approach to learning with instance-dependent label noise. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=ecH2FKaARUp
20. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. pp. 5132–5143. PMLR (2020)
21. Kim, S., Shin, W., Jang, S., Song, H., Yun, S.Y.: Fedrn: Exploiting k-reliable neighbors towards robust federated learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 972–981 (2022)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=HJgExaVtwr
24. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems **2**, 429–450 (2020)
25. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189 (2019)
26. Li, X., Jiang, M., Zhang, X., Kamp, M., Dou, Q.: Fedbn: Federated learning on non-iid features via local batch normalization. arXiv preprint arXiv:2102.07623 (2021)
27. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. Advances in neural information processing systems **33**, 20331–20342 (2020)
28. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. IEEE Transactions on pattern analysis and machine intelligence **38**(3), 447–461 (2015)
29. Liu, Y.: Understanding instance-level label noise: Disparate impacts and treatments. In: International Conference on Machine Learning. pp. 6725–6735. PMLR (2021)
30. Liu, Y., Guo, H.: Peer loss functions: Learning from noisy labels without knowing noise rates. In: Proceedings of the 37th International Conference on Machine Learning. ICML '20 (2020)

31. Liu, Y., **Jialu Wang**: Can less be more? when increasing-to-balancing label noise rates considered beneficial. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 17467–17479. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper/2021/file/91e50fe1e39af2869d3336eaaeebdb43-Paper.pdf
32. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
33. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE symposium on security and privacy (SP). pp. 691–706. IEEE (2019)
34. Menon, A., Van Rooyen, B., Ong, C.S., Williamson, B.: Learning from corrupted binary labels via class-probability estimation. In: International Conference on Machine Learning. pp. 125–134 (2015)
35. Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A.: Learning with noisy labels. In: Advances in neural information processing systems. pp. 1196–1204 (2013)
36. Pan, X., Zhang, M., Ji, S., Yang, M.: Privacy risks of general-purpose language models. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 1314–1331. IEEE (2020)
37. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1944–1952 (2017)
38. Qin, Z., Yao, L., Chen, D., Li, Y., Ding, B., Cheng, M.: Revisiting personalized federated learning: Robustness against backdoor attacks. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 4743–4755 (2023)
39. Scott, C.: A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In: AISTATS (2015)
40. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1310–1321 (2015)
41. Song, H., Kim, M., Park, D., Lee, J.G.: How does early stopping help generalization against label noise? arXiv preprint arXiv:1911.08059 (2019)
42. **Jialu Wang**, Liu, Y., Levy, C.: Fair classification with group-dependent label noise. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. p. 526–536. FAccT '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442188.3445915, https://doi.org/10.1145/3442188.3445915
43. Tuor, T., Wang, S., Ko, B.J., Liu, C., Leung, K.K.: Overcoming noisy and irrelevant data in federated learning. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 5020–5027. IEEE (2021)
44. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Open-set recognition: A good closed-set classifier is all you need? arXiv preprint arXiv:2110.06207 (2021)
45. Wang, J., Liu, Y., Levy, C.: Fair classification with group-dependent label noise. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. pp. 526–536 (2021)
46. Wei, J., Liu, H., Liu, T., Niu, G., Liu, Y.: To smooth or not? when label smoothing meets noisy labels. In: ICML (2022)

47. Wei, J., Liu, Y.: When optimizing f-divergence is robust with label noise. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=WesiCoRVQ15
48. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with noisy labels revisited: A study using real-world human annotations. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=TBWA6PLJZQm
49. Wei, J., Zhu, Z., Luo, T., Amid, E., Kumar, A., Liu, Y.: To aggregate or not? learning with separate noisy labels. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2023)
50. Wu, N., Yu, L., Jiang, X., Cheng, K.T., Yan, Z.: Fednoro: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity. arXiv preprint arXiv:2305.05230 (2023)
51. Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., Chang, Y.: Robust early-learning: Hindering the memorization of noisy labels. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=Eql5b1_hTE4
52. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? Advances in Neural Information Processing Systems **32** (2019)
53. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2691–2699 (2015)
54. Xu, J., Chen, Z., Quek, T.Q., Chong, K.F.E.: Fedcorr: Multi-stage federated learning for label noise correction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10184–10193 (2022)
55. Yang, S., Park, H., Byun, J., Kim, C.: Robust federated learning with noisy labels. IEEE Intelligent Systems **37**(2), 35–43 (2022)
56. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=r1Ddp1-Rb
57. Zhang, J., Sheng, V.S., Li, T., Wu, X.: Improving crowdsourced label quality using noise correction. IEEE transactions on neural networks and learning systems **29**(5), 1675–1688 (2017)
58. Zhang, L., Luo, Y., Bai, Y., Du, B., Duan, L.Y.: Federated learning for non-iid data via unified feature learning and optimization objective alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4420–4428 (2021)
59. Zhang, Y., Niu, G., Sugiyama, M.: Learning noise transition matrix from only noisy labels via total variation regularization. arXiv preprint arXiv:2102.02414 (2021)
60. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in neural information processing systems. pp. 8778–8788 (2018)
61. Zhao, B., Mopuri, K.R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)
62. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)
63. Zhu, Z., Liu, T., Liu, Y.: A second-order approach to learning with instance-dependent label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10113–10123 (2021)

64. Zhu, Z., Song, Y., Liu, Y.: Clusterability as an alternative to anchor points when learning with noisy labels. In: International Conference on Machine Learning. pp. 12912–12923. PMLR (2021)
65. Zhu, Z., Wang, J., Cheng, H., Liu, Y.: Unmasking and improving data credibility: A study with datasets for training harmless language models. arXiv preprint arXiv:2311.11202 (2023)
66. Zhu, Z., Wang, J., Cheng, H., Liu, Y.: Unmasking and improving data credibility: A study with datasets for training harmless language models. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=6bcAD6g688
67. Zhu, Z., Wang, J., Liu, Y.: Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. arXiv preprint arXiv:2202.01273 (2022)

Supplementary Materials

Roadmap The appendix is composed as follows. Section 7 presents all the notations and their meaning we use in this paper. Section 8 gives the proof and analysis omitted in the main paper. Section 9 introduces the implementation details of the experiments and how to apply the centralized training methods to FL. Section 10 visualizes the data distribution of CIFAR-10 across 100 clients and shows how difficult the openset noisy label problem is. Section 11 discusses the connection between FedDPContand personal federated learning. Section 12 shows the experiment results with more details that are not given in the main paper due to the page limit.

7 Notation Table

Table 5: Table of notations used in the paper

Notation	Explanation
η	Noisy ratio
C	Total number of clients
c	Client c in federated learning
$\tilde{Y}$	Random variables for the noisy label
$\hat{Y}$	Random variables for the output of the model
K	Number of classes in $\mathcal{Y}$
T	Transition matrix
$\mathbb{P}$	The probability
$\mathbb{E}$	The expectation
λ	Federated fraction to control the number of clients in every round
α_g	The global step size on the server side
α_l	The local learning rate on the client side
L, ℓ	The loss, the loss function
R	The global communication round
E	The local updating round
T_{DP}	Differential privacy transition matrix
T	Transition matrix
$\mathcal{A}$	The label communication algorithm
$\check{Y}_c$	Labels protected by differential privacy
$\tilde{\boldsymbol{p}}_n^c$	one-hot encoding of $\tilde{y}_n^c$
$\check{\boldsymbol{p}}$	The posterior label distribution after differential privacy corruption
θ, θ_c^r	The model parameters, the model parameters of client c at r-th round
e_1, e_2	The noisy ratio of class 1 and 2 of the global dataset in binary classification
e_1^k, e_2^k	The noisy ratio of class 1 and 2 of the local client k in binary classification
m_1	The number of samples which are wrongly labeled from 1 to 2 in binary classification
m_2	The number of samples which are wrongly labeled from 2 to 1 in binary classification
$\Delta_c^{(r)}$	The variation of model parameters in r-th round of the client c
X, Y	Random variables for the feature and label
$\mathcal{X}, \mathcal{Y}$	The space of X, Y
$\boldsymbol{f}_c, \boldsymbol{f}_g$	The client model, The global model
N, N_c	Total number of samples, number of samples in client c
(x_n^c, y_n^c)	The n-th example in the client c
$D_c := \{(x_n^c, y_n^c)\}_{n\in[N_c]}$	Dataset of client c
$D := \{(x_n, y_n)\}_{n\in[N]}$	Dataset
$I_k := \{c \vert \mathbb{1}_{c,k} = 1, c \in [C]\}$	The vector indicating whether client c can access class k or not

8 Proofs and Analyses

In this section, we present all the proofs of the theorems.

8.1 Proof of Theorem 1

Proof. Denote by $\mathcal{A}$ the label communication algorithm, where the input is y and the output is y_{DP}. Then after flipping the label y according to the noise transition matrix T, we have

$$\mathbb{P}(\mathcal{A}(y) = y_{\mathsf{DP}}) = \begin{cases} \frac{e^\epsilon}{e^\epsilon + K - 1}, \text{if } y_{\mathsf{DP}} = y, \\ \frac{1}{e^\epsilon + K - 1}, \text{if } y_{\mathsf{DP}} \neq y. \end{cases}$$

Accordingly, for another label y', we have

$$\mathbb{P}(\mathcal{A}(y') = y_{\mathsf{DP}}) = \begin{cases} \frac{e^\epsilon}{e^\epsilon + K - 1}, \text{if } y_{\mathsf{DP}} = y', \\ \frac{1}{e^\epsilon + K - 1}, \text{if } y_{\mathsf{DP}} \neq y'. \end{cases}$$

Then the quotient of two probabilities can be upper bounded by

$$\frac{\mathbb{P}(\mathcal{A}(y) = y_{\mathsf{DP}})}{\mathbb{P}(\mathcal{A}(y') = y_{\mathsf{DP}})} \leq e^\epsilon.$$

With Definition 2, we know the above equation is exactly the definition of ϵ-labelDP, i.e., the label communication algorithm is ϵ-labelDP.

□

8.2 Proof of Theorem 2

Proof. The centralized private loss on $\mathcal{D}$ is

$$\mathbb{E}_{\mathcal{D}}[\ell_{\mathrm{PL}}(f(X), \widetilde{Y})] = \mathbb{E}_{\mathcal{D}}\left[\ell(f(X), \widetilde{Y}) - \beta \cdot \mathbb{E}_{\mathcal{D}_{\widetilde{Y}'|\widetilde{D}}}[\ell(f(X), \widetilde{Y}')]\right],$$

where $\widetilde{Y}'$ is the random variable whose distribution is the noisy label distribution. For each client c, the local FedDPCont loss is

$$\mathbb{E}_{\mathcal{D}_c}[\ell_{\mathrm{FedDPCont}}(f(X_c), \widetilde{Y}_c)] = \mathbb{E}_{\mathcal{D}_c}\left[\ell(f(X_c), \widetilde{Y}_c) - \beta \cdot \mathbb{E}_{\mathcal{D}_{\widetilde{Y}'|\widetilde{D}}}[\ell(f(X_c), \widetilde{Y}')]\right],$$

.

Denote by $\mathbb{P}(\mathcal{D}_c|\mathcal{D})$ the probability of drawing a data point from client c. We have

$$\sum_{c \in [C]} \mathbb{P}(\mathcal{D}_c|\mathcal{D}) = 1.$$

Then

$$\begin{aligned}
&\sum_{c\in[C]} \mathbb{P}(\mathcal{D}_c|\mathcal{D})\mathbb{E}_{\mathcal{D}_c}[\ell_{\text{FedDPCont}}(f(X_c),\widetilde{Y}_c)] \\
=&\sum_{c\in[C]} \mathbb{P}(\mathcal{D}_c|\mathcal{D})\mathbb{E}_{\mathcal{D}_c}\left[\ell(f(X_c),\widetilde{Y}_c)-\beta\cdot\mathbb{E}_{\mathcal{D}_{\widetilde{Y}'|\widetilde{D}}}[\ell(f(X_c),\widetilde{Y}')]\right] \\
=&\mathbb{E}_{\mathcal{D}}\left[\ell(f(X),\widetilde{Y})-\beta\cdot\mathbb{E}_{\mathcal{D}_{\widetilde{Y}'|\widetilde{D}}}[\ell(f(X),\widetilde{Y}')]\right]=\mathbb{E}_{\mathcal{D}}[\ell_{\text{peer}}(f(X),\widetilde{Y})].
\end{aligned}$$

Each round may include multiple epochs. Suppose there are t local epochs. The variation of model parameters in the r-th round of the local training in client c can be decomposed by

$$\begin{aligned}
\Delta_c^{(r)} :=&\theta_c^{(r+1)}-\theta_c^{(r)}=\theta_c^{(r+1,t)}-\theta_c^{(r+1,t-1)}+\theta_c^{(r+1,t-1)}+\cdots-\theta_c^{(r,1)} \\
=&\frac{\partial\mathbb{E}_{\mathcal{D}_c}[\ell_{\text{FedDPCont}}(f(X_c),\widetilde{Y}_c;\theta_c)]}{\partial\theta_c}\bigg|_{\theta=\theta_c^{(r+1,t-1)}}+\cdots \\
&+\frac{\partial\mathbb{E}_{\mathcal{D}_c}[\ell_{\text{FedDPCont}}(f(X_c),\widetilde{Y}_c;\theta_c)]}{\partial\theta_c}\bigg|_{\theta=\theta_c^{(r+1,1)}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\sum_{c\in[C]} \mathbb{P}(\mathcal{D}_c|\mathcal{D})\Delta_c^{(r)} =&\frac{\partial\sum_{c\in[C]}\mathbb{P}(\mathcal{D}_c|\mathcal{D})\mathbb{E}_{\mathcal{D}_c}[\ell_{\text{FedDPCont}}(f(X_c),\widetilde{Y}_c;\theta_c^{(r+1,t-1)})]}{\partial\theta_c} \\
&+\cdots+\frac{\partial\sum_{c\in[C]}\mathbb{P}(\mathcal{D}_c|\mathcal{D})\mathbb{E}_{\mathcal{D}_c}[\ell_{\text{FedDPCont}}(f(X_c),\widetilde{Y}_c;\theta_c^{(r+1,1)})]}{\partial\theta_c} \\
=&\frac{\partial\mathbb{E}_{\mathcal{D}}[\ell_{\text{PL}}(f(X),\widetilde{Y};\theta^{(r+1,t-1)})]}{\partial\theta}+\cdots+\frac{\partial\mathbb{E}_{\mathcal{D}}[\ell_{\text{PL}}(f(X),\widetilde{Y};\theta^{(r+1,1)})]}{\partial\theta} \\
=&\Delta^{(r)}.
\end{aligned}$$

□

8.3 Details about the Necessity of Using a Global Private Label

To be more concrete, in [30], for each example $(x_n,\tilde{y}_n)$, the private loss defines as (an equivalent form):

$$\ell_{\text{PL}}(\boldsymbol{f}(x_n),\tilde{y}_n) := \ell(\boldsymbol{f}(x_n),\tilde{y}_n)-\ell(\boldsymbol{f}(x_n),\tilde{y}_{n'}), \tag{3}$$

where $\tilde{y}_{n'}$ is a randomly sampled constrastive label. Later as a follow-up work [7], ℓ_{CORES} was proposed as a more stable version of ℓ_{PL} which has the same expectation as ℓ_{PL}:

$$\ell_{\text{CORES}}(\boldsymbol{f}(x_n),\tilde{y}_n)=\ell(\boldsymbol{f}(x_n),\tilde{y}_n)-\mathbb{E}_{\mathcal{D}_{\tilde{Y}|\tilde{D}}}[\ell(\boldsymbol{f}(x_n),\tilde{Y}], \tag{4}$$

where $\mathcal{D}_{\tilde{Y}|\tilde{D}}$ is the distribution of $\tilde{Y}$ given dataset $\tilde{\mathcal{D}}$. Private loss and ℓ_{CORES} have strong consistency guarantees. Consider a binary classification problem and let $e_1 := \mathbb{P}(\tilde{Y}=2|Y=1)$ and $e_2=\mathbb{P}(\tilde{Y}=1|Y=2)$. Then it was proved in [30] the following robustness of peer loss:

Proposition 1 (Robustness of peer loss [30]). *Peer loss is invariant to label noise:*

$$\mathbb{E}_{\tilde{\mathcal{D}}}[\ell_{PL}(f(X), \tilde{Y})] = (1 - e_1 - e_2) \cdot \mathbb{E}_{\mathcal{D}}[\ell_{PL}(f(X), Y)].$$

Moreover, when $\mathbb{P}(Y = 1) = 0.5$ *and* ℓ *is the 0-1 loss, minimizing peer loss on noisy distribution* $\tilde{\mathcal{D}}$ *is equivalent to minimizing 0-1 loss on clean distribution* $\mathcal{D}$*.*

Can we then follow the above idea and implement either ℓ_{PL} or ℓ_{CORES} by requiring each client to sample the "contrastive labe" $\tilde{y}_{n'}$ locally? Unfortunately, the answer is no. There are two **technical challenges**:

First, sampling contrastive labels locally leads to ***wrong*** *results.* A local sampling for the private label will lead to a distribution that does not capture the global one on $\mathbb{P}(\tilde{Y})$, then challenge the theoretical guarantees of the existing results. To see this, we consider a binary classification problem. Assume that we have two clients $c = 1$ and $c = 2$, where client 1 can only access noisy labels 1 and client 2 only accesses noisy labels 2, respectively. Suppose the number of data points in each class (globally) is $N_1 = N_2 = N/2$. If there are m_1 samples that are wrongly labeled from $Y = 1$ to $\tilde{Y} = 2$ and m_2 samples that are wrongly labeled from $Y = 2$ to $\tilde{Y} = 1$, respectively, we can know the global noisy ratios are $e_1 = \mathbb{P}(\tilde{Y} = 2|Y = 1) = 2m_1/N$ and $e_2 = \mathbb{P}(\tilde{Y} = 1|Y = 2) = 2m_2/N$, respectively. For centralized training, we know from Proposition 1 that there is an invariant property. However, due to the openset, the locally noisy ratio differs from the globally noisy ratio and the invariant property is **broken**. Specifically, the local noisy ratios are $e_1^1 = \mathbb{P}(\tilde{Y}_1 = 2|Y_1 = 1) = 0$ and $e_2^1 = \mathbb{P}(\tilde{Y}_1 = 1|Y_1 = 2) = 1$ where Y_1 is the label and $\tilde{Y}_1$ is the corrupted label in client 1. Then the invariant property in Proposition 1 becomes

$$\mathbb{E}_{\tilde{\mathcal{D}}_1}[\ell_{\mathrm{PL}}(f(X), \tilde{Y})] = (1 - e_1^1 - e_2^1) = 0,$$

which is a constant for any model f. Therefore, peer labels need to be redesigned in FL with openset noisy labels.

Second, there are privacy concerns in redesigning contrastive labels. Intuitively, since we know local sampling fails, the global information is inevitable in redesigning peer labels. Therefore, the privacy issues need to be addressed in label communications.

9 Implementation Details

9.1 Platform and Programming Environment

We train our model on NVIDIA RTX A5000 server with torch and torchvision 1.10 and 0.11, respectively. The details of the baseline methods are as follows.

9.2 Loss correction

We apply FedAvg in the first 150 rounds to make the weight stable. At the 150th round, the transition matrix of every client will be estimated according

to the confidential score of 95%. The predicted label whose confidential score is over 95% is considered as the ground truth so that we can get every transition matrix of every client. We apply loss correction in the rest 150 rounds according to Equation 1.

9.3 Co-teaching

Co-teaching uses two same networks to distinguish the noisy data and the clean data. Similarly, we initialize two same networks when the client initializes and update the two clients in the same way as the original co-teaching network. The server also keeps two models. In every communication round, the weights of the two models will average correspondingly.

9.4 T-revision

T-revision consists of three steps: estimation of T, loss correction, and T-revision. In the first 20 communication rounds, the selected clients update the weight at every communication round and all the clients estimate T_c. After the 20th round, the selected clients at every communication apply forward loss correction for another 140 rounds. After the 160th round, we apply T-revision.

9.5 DivideMix

DivideMix uses two same networks to distinguish the noisy label. One network is used to assign the pseudo label, the other network is used to the classification. The pseudo label is generated by a Gaussian mixture process. In addition, DivideMix uses mix-up data augmentation to boost performance. In FL paradigm, every client will maintain two clients and do the same operation as the centralized training in DivideMix.

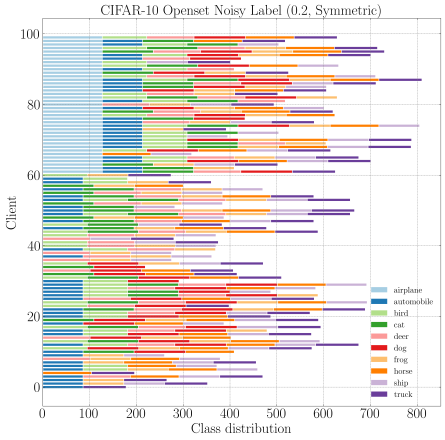
For the other baseline methods, we follow the original settings in their papers.

10 Openset Noisy Label Visualization

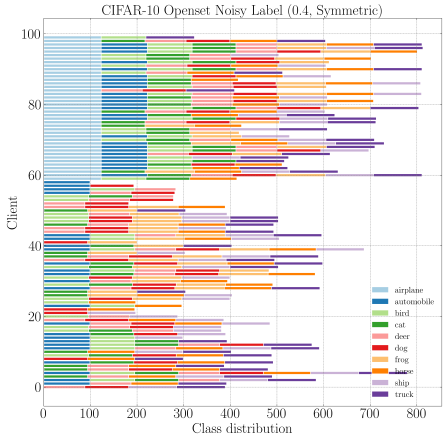
We visualize the data distribution across the 100 clients in CIFAR-10 when the noise types are both symmetric and random in Fig. 3 and 4, respectively. It should be noted that the observed label space is corrupted and does not cover the whole label space. For each client, the true label is unknown. The data distribution is uniform when the noise is symmetric while the the data distribution is chaotic when the noise is random.

11 The Relationship to Personalized Federated Learning

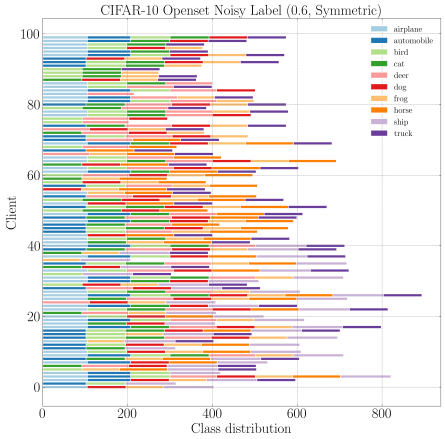
It would be interesting to investigate the robustness of noisy label in personalized FL (pFL) because pFL provides more flexibility for the local model [6,38]. Moreover, FedDPCont is a loss-based method which is independent from the model



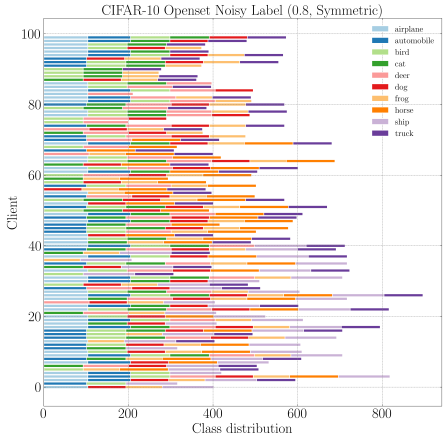
(a) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.2** and the noise type is **symmetric**



(b) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.4** and the noise type is **symmetric**



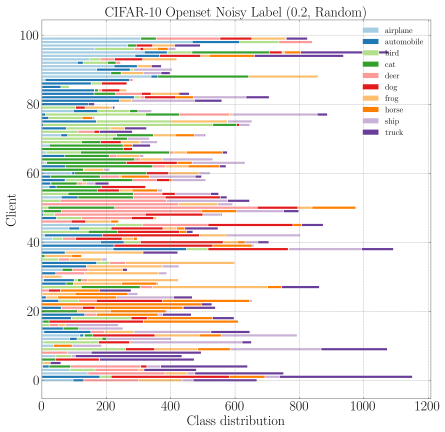
(c) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.6** and the noise type is **symmetric**



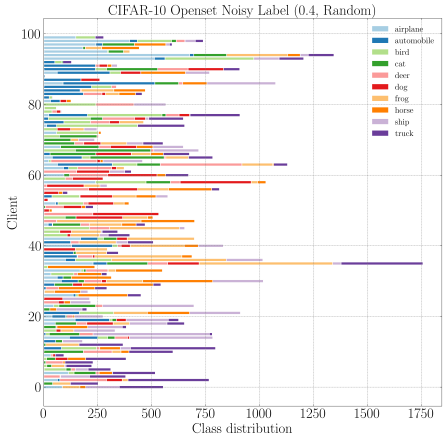
(d) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.8** and the noise type is **symmetric**

Fig. 3: The data distribution of CIFAR-10 across 100 clients under different noise ratio when the noise is **symmetric**.

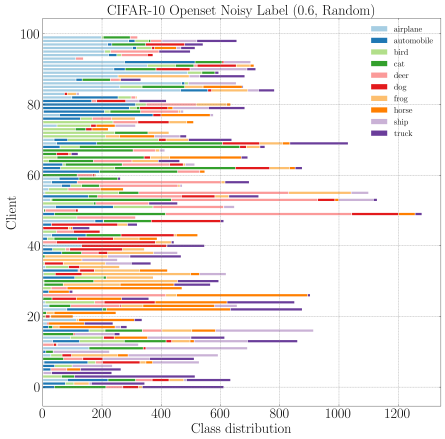
architecture. The combination of FedDPCont and pFL is also worth investigation. In particular, we hypothesize that by only partially sharing the contrastive label distribution, our noise correction loss will induce a personalized correction at local clients, instead of “over-correcting" using a global one.



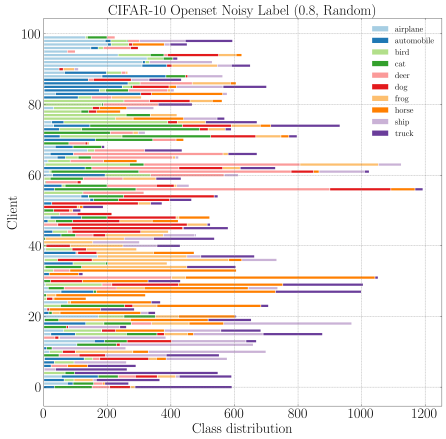
(a) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.2** and the noise type is **random**



(b) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.4** and the noise type is **random**



(c) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.6** and the noise type is **random**



(d) The data distribution across the 100 clients of CIFAR-10 dataset when the noise ratio is **0.8** and the noise type is **random**

Fig. 4: The data distribution of CIFAR-10 across 100 clients under different noise ratio when the noise is **random**.

12 Experiment Results

12.1 More Results

Due to the page limit in the main text, we report the full results on CIFAR-100, CIFAR-N and Clothing1M in Table 6 and 7.

Table 6: The performance (the best accuracy) of all the methods on CIFAR-100. FedDPCont is always the best method.

Dataset	Methods	Symmetric				Random			
		0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
CIFAR-100	FedAvg	$47.78_{\pm 0.50}$	$32.63_{\pm 0.27}$	$20.32_{\pm 0.51}$	$10.62_{\pm 0.26}$	$47.75_{\pm 0.29}$	$31.06_{\pm 0.79}$	$20.14_{\pm 0.32}$	$9.71_{\pm 0.43}$
	LC	$48.92_{\pm 0.42}$	$33.15_{\pm 0.23}$	$20.39_{\pm 0.36}$	$10.43_{\pm 0.45}$	$49.03_{\pm 0.17}$	$32.67_{\pm 0.75}$	$19.78_{\pm 0.67}$	$10.13_{\pm 0.36}$
	FedProx	$32.14_{\pm 0.27}$	$24.68_{\pm 0.11}$	$16.52_{\pm 0.77}$	$8.85_{\pm 0.60}$	$31.77_{\pm 0.30}$	$25.03_{\pm 0.47}$	$17.16_{\pm 0.64}$	$8.84_{\pm 0.50}$
	Co-teaching	$41.15_{\pm 0.28}$	$29.81_{\pm 0.72}$	$18.01_{\pm 0.28}$	$8.73_{\pm 1.08}$	$40.55_{\pm 1.79}$	$28.51_{\pm 1.41}$	$18.47_{\pm 1.95}$	$6.56_{\pm 1.38}$
	T-revision	$48.21_{\pm 0.56}$	$31.35_{\pm 0.46}$	$17.41_{\pm 0.22}$	$7.79_{\pm 0.28}$	$48.24_{\pm 0.47}$	$30.91_{\pm 0.55}$	$16.95_{\pm 0.78}$	$7.46_{\pm 0.20}$
	FedDyn	$31.73_{\pm 0.79}$	$23.35_{\pm 0.23}$	$15.53_{\pm 0.21}$	$7.82_{\pm 0.04}$	$32.22_{\pm 0.35}$	$23.83_{\pm 0.42}$	$16.27_{\pm 0.59}$	$7.86_{\pm 0.10}$
	FedBN	$40.71_{\pm 1.19}$	$25.61_{\pm 0.53}$	$14.52_{\pm 0.18}$	$6.64_{\pm 0.32}$	$38.96_{\pm 0.86}$	$24.54_{\pm 0.86}$	$13.52_{\pm 0.73}$	$6.63_{\pm 0.17}$
	Scaffold	$31.56_{\pm 0.20}$	$24.85_{\pm 0.38}$	$14.42_{\pm 0.93}$	$2.10_{\pm 0.35}$	$28.49_{\pm 0.75}$	$21.74_{\pm 0.48}$	$11.19_{\pm 1.23}$	$1.97_{\pm 0.44}$
	FedCorr	$57.65_{\pm 0.37}$	$44.48_{\pm 1.04}$	$27.37_{\pm 0.25}$	$12.09_{\pm 0.23}$	$55.04_{\pm 0.79}$	$43.71_{\pm 2.05}$	$25.91_{\pm 0.14}$	$11.96_{\pm 0.71}$
	FedRN	$46.84_{\pm 0.71}$	$34.40_{\pm 0.68}$	$22.63_{\pm 0.44}$	$11.30_{\pm 0.13}$	$46.24_{\pm 0.66}$	$33.48_{\pm 1.30}$	$21.33_{\pm 0.59}$	$10.41_{\pm 0.14}$
	RoFL	$40.23_{\pm 1.33}$	$32.86_{\pm 2.68}$	$21.90_{\pm 0.03}$	$13.24_{\pm 0.10}$	$39.44_{\pm 0.96}$	$30.10_{\pm 0.26}$	$21.53_{\pm 1.52}$	$11.46_{\pm 0.03}$
	FedNoRo	$16.85_{\pm 0.35}$	$9.80_{\pm 0.32}$	$5.62_{\pm 0.10}$	$2.04_{\pm 0.48}$	$15.56_{\pm 0.47}$	$7.47_{\pm 0.44}$	$5.17_{\pm 0.24}$	$1.29_{\pm 0.06}$
	RHFL	$16.56_{\pm 0.72}$	$11.92_{\pm 0.75}$	$5.09_{\pm 0.09}$	$3.58_{\pm 0.57}$	$14.99_{\pm 0.45}$	$9.34_{\pm 0.41}$	$4.30_{\pm 0.62}$	$1.75_{\pm 0.59}$
	FedDPCont	$\mathbf{58.55}_{\pm 0.65}$	$\mathbf{46.65}_{\pm 0.53}$	$\mathbf{29.27}_{\pm 0.79}$	$\mathbf{14.04}_{\pm 0.96}$	$\mathbf{57.73}_{\pm 0.36}$	$\mathbf{45.43}_{\pm 0.72}$	$\mathbf{27.35}_{\pm 0.72}$	$\mathbf{12.64}_{\pm 0.43}$

Table 7: The performance (the best accuracy) of all methods on CIFAR-N and Clothing-1M.

Datasets	CIFAR-10			CIFAR-100	Clothing-1M
Methods	Worst	Random	Aggregate	Fine	1M Noisy Training
FedAvg	$46.55_{\pm 7.82}$	$59.69_{\pm 4.88}$	$66.41_{\pm 6.52}$	$22.65_{\pm 2.29}$	70.27
LC	$46.67_{\pm 8.21}$	$59.27_{\pm 5.72}$	$67.27_{\pm 4.76}$	$22.59_{\pm 1.66}$	70.05
FedProx	$58.47_{\pm 0.97}$	$69.35_{\pm 0.62}$	$74.48_{\pm 1.00}$	$35.33_{\pm 0.35}$	65.96
Co-teaching	$24.80_{\pm 2.27}$	$47.34_{\pm 21.05}$	$62.04_{\pm 11.26}$	$17.83_{\pm 0.39}$	40.33
T-revision	$57.85_{\pm 19.44}$	$55.06_{\pm 8.40}$	$63.40_{\pm 9.99}$	$22.18_{\pm 1.44}$	66.95
FedBN	$63.07_{\pm 3.29}$	$73.02_{\pm 1.45}$	$77.55_{\pm 2.16}$	$37.59_{\pm 0.61}$	67.43
FedCorr	$49.36_{\pm 4.66}$	$58.18_{\pm 9.97}$	$64.06_{\pm 7.58}$	$50.42_{\pm 0.37}$	69.55
FedRN	$49.96_{\pm 0.86}$	$59.70_{\pm 0.69}$	$64.08_{\pm 1.00}$	$43.31_{\pm 0.20}$	62.78
FedNoRo	$33.33_{\pm 1.30}$	$44.48_{\pm 2.25}$	$47.13_{\pm 1.51}$	$17.01_{\pm 0.56}$	41.11
RHFL	$34.15_{\pm 0.54}$	$44.23_{\pm 2.38}$	$45.69_{\pm 1.20}$	$15.49_{\pm 0.68}$	39.93
RoFL	$17.75_{\pm 0.89}$	$20.58_{\pm 0.38}$	$16.19_{\pm 0.41}$	$35.17_{\pm 2.21}$	-
FedDPCont	$\mathbf{63.50}_{\pm 5.63}$	$\mathbf{73.68}_{\pm 4.35}$	$\mathbf{81.86}_{\pm 1.09}$	$\mathbf{52.60}_{\pm 1.91}$	**70.88**

12.2 DivideMix Details

Compared with DivideMix, FedDPCont is a lightweight method. Due to the mix-up augmentation method and dual-model architecture design, DivideMix needs more time to converge. We compare the performance of DivideMix and FedDPCont in terms of the epoch and the training time. All the experiments are done on a server with an AMD EPYC 7513 32-Core Processor and RTX A5000 NVIDIA GPU to guarantee the training time is calculated fairly. The results on benchmark and real-world datasets are given in Table 8 and 9.

We can find that FedDPCont needs much less time than DivideMix in all cases and outperforms on all CIFAR-10 datasets for both benchmark and real-world cases. FedDPCont depends heavily on the estimation of the distribution of the dataset from label communication as given in Section 4.2. When the data

belonging to each class is fewer or the noisy ratio is higher, the difficulty of precise estimation becomes much larger. Compared with FedDPCont, DivideMix uses another model to generate the pseudo label so that the performance can be less sensitive to the heterogeneity but will be much slower. In the practical usage, FedDPCont is a reliable choice in terms of speed and performance.

Table 8: Comparison of DivideMix and FedDPCont in terms of time and number of epochs on benchmark dataset. R stands for random noise in Section 5.2. Compared with DivideMix, FedDPCont is lightweight and can produce relatively reliable results.

Dataset	Methods	0.2	0.4	0.6	0.8	Epochs	Time (hr)
CIFAR-10	DivideMix	$79.28_{\pm 0.33}$	$65.62_{\pm 4.48}$	$53.28_{\pm 2.16}$	$20.70_{\pm 5.00}$	300	22.33
	FedDPCont	$\mathbf{84.77}_{\pm 0.12}$	$\mathbf{75.75}_{\pm 1.96}$	$\mathbf{55.50}_{\pm 1.33}$	$\mathbf{24.64}_{\pm 0.55}$	300	4.17
CIFAR-10 (R)	DivideMix	$70.31_{\pm 2.19}$	$59.24_{\pm 1.90}$	$48.91_{\pm 4.59}$	$24.29_{\pm 2.19}$	300	27.67
	FedDPCont	$\mathbf{82.15}_{\pm 0.24}$	$\mathbf{72.69}_{\pm 1.57}$	$\mathbf{54.06}_{\pm 1.38}$	$\mathbf{27.55}_{\pm 1.49}$	300	3.90
CIFAR-100	DivideMix	$58.31_{\pm 0.47}$	$46.62_{\pm 0.37}$	$\mathbf{31.90}_{\pm 0.85}$	$\mathbf{19.87}_{\pm 0.52}$	300	30.33
	FedDPCont	$\mathbf{58.55}_{\pm 0.65}$	$\mathbf{46.65}_{\pm 0.53}$	$29.27_{\pm 0.79}$	$14.04_{\pm 0.96}$	300	3.87
CIFAR-100 (R)	DivideMix	$57.31_{\pm 0.08}$	$\mathbf{45.60}_{\pm 0.51}$	$\mathbf{30.95}_{\pm 0.57}$	$\mathbf{19.40}_{\pm 0.53}$	300	29.33
	FedDPCont	$\mathbf{57.73}_{\pm 0.36}$	$45.43_{\pm 0.72}$	$27.35_{\pm 0.72}$	$12.64_{\pm 0.43}$	300	4.28

Table 9: Comparison of DivideMix and FedDPCont in terms of time and number of epochs on the noisy real-world dataset. Compared with DivideMix, FedDPCont is lightweight and can produce relatively reliable results.

Dataset	Methods	Accuracy	Epochs	Time (hr)
CIFAR-10-N-Worst	DivideMix	$59.50_{\pm 5.90}$	300	33.00
	FedDPCont	$\mathbf{63.50}_{\pm 5.63}$	300	3.76
CIFAR-10-N-Random	DivideMix	$66.45_{\pm 2.69}$	300	21.50
	FedDPCont	$\mathbf{73.68}_{\pm 4.35}$	300	3.43
CIFAR-10-N-Aggregate	DivideMix	$71.98_{\pm 2.27}$	300	25.50
	FedDPCont	$\mathbf{81.86}_{\pm 1.09}$	300	3.83
CIFAR-100-N	DivideMix	$45.66_{\pm 0.15}$	300	13.85
	FedDPCont	$\mathbf{52.60}_{\pm 1.91}$	300	2.63

12.3 More Results about the Effect of Differential Privacy Level

In addition to the results in Table 4, we also do a comprehensive experiment for the influence of ϵ on the performance on different dataset with different noise ratio. The results on CIFAR-10, CIFAR-100 and Clothing1M are given in Table 10, 11 and 12, where $p = \frac{e^\epsilon}{e^\epsilon + K - 1}, \epsilon > 0$.

Table 10: The performance of FedDPCont on different differential privacy levels. The number indicates the noise ratio. R indicates the random noise. The last three columns are the results from the CIFAR-10 subset in CIFAR-N. The last row in gray is the number we report in the main paper.

ϵ	p	0.2	0.4	0.6	0.8	0.2 (R)	0.4 (R)	0.6 (R)	0.8 (R)	Worst	Aggre	Random
0.01	0.1	**85.03**$_{\pm 0.35}$	75.19$_{\pm 1.00}$	**56.01**$_{\pm 0.03}$	24.31$_{\pm 0.45}$	**82.20**$_{\pm 0.62}$	72.12$_{\pm 1.08}$	53.79$_{\pm 0.92}$	25.10$_{\pm 0.66}$	63.71$_{\pm 3.31}$	73.21$_{\pm 2.22}$	81.11$_{\pm 0.98}$
0.81	0.2	83.43$_{\pm 0.04}$	75.23$_{\pm 0.05}$	55.85$_{\pm 0.42}$	24.07$_{\pm 0.10}$	81.79$_{\pm 0.76}$	**72.99**$_{\pm 1.01}$	**54.11**$_{\pm 0.63}$	25.03$_{\pm 1.71}$	**64.59**$_{\pm 2.70}$	73.04$_{\pm 3.43}$	81.79$_{\pm 0.79}$
-	-	84.77$_{\pm 0.12}$	**75.75**$_{\pm 1.96}$	55.50$_{\pm 1.33}$	**24.64**$_{\pm 0.55}$	82.15$_{\pm 0.24}$	72.69$_{\pm 1.57}$	54.06$_{\pm 1.38}$	**27.55**$_{\pm 1.49}$	63.50$_{\pm 5.63}$	**73.68**$_{\pm 4.35}$	**81.86**$_{\pm 1.09}$

Table 11: The performance of FedDPCont on different differential privacy levels. The number indicates the noise ratio. R indicates the random noise. The last columns are the results from the CIFAR-100 subset in CIFAR-N. The last row in gray is the number we report in the main paper.

ϵ	p	0.2	0.4	0.6	0.8	0.2 (R)	0.4 (R)	0.6 (R)	0.8 (R)	Fine
0.01	0.01	58.05$_{\pm 0.98}$	46.71$_{\pm 0.29}$	29.01$_{\pm 0.24}$	13.51$_{\pm 0.01}$	56.79$_{\pm 0.33}$	**45.43**$_{\pm 0.63}$	27.75$_{\pm 0.42}$	12.70$_{\pm 0.87}$	**53.06**$_{\pm 0.02}$
3.21	0.2	58.17$_{\pm 0.93}$	**46.86**$_{\pm 0.22}$	28.93$_{\pm 0.26}$	13.62$_{\pm 0.94}$	56.44$_{\pm 0.51}$	44.31$_{\pm 0.52}$	**27.89**$_{\pm 0.99}$	**13.01**$_{\pm 0.40}$	52.11$_{\pm 0.53}$
-	-	**58.55**$_{\pm 0.65}$	46.65$_{\pm 0.53}$	**29.27**$_{\pm 0.79}$	**14.04**$_{\pm 0.96}$	**57.73**$_{\pm 0.36}$	45.43$_{\pm 0.72}$	27.35$_{\pm 0.72}$	12.64$_{\pm 0.43}$	52.60$_{\pm 1.91}$

Table 12: The performance of FedDPCont on different differential privacy levels on Clothin1M. The last column in gray is the number we report in the main paper.

$\epsilon = 0.01, p = 0.071$	$\epsilon = 1.18, p = 0.2$	Main Text
70.80	70.79	70.88