FEDERATED LEARNING WITH HETEROGENEOUS LABEL NOISE: A DUAL STRUCTURE APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

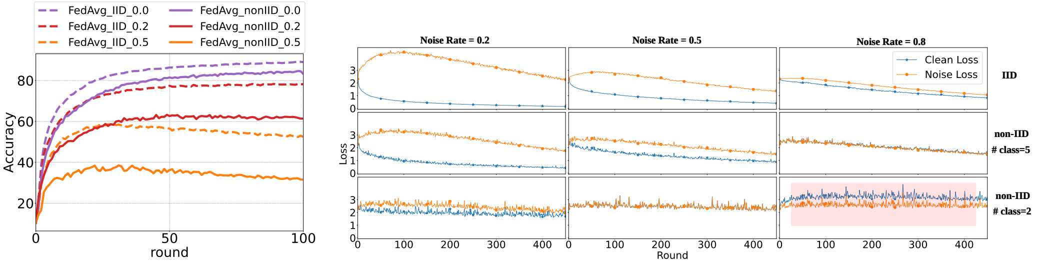
The performance of federated learning relies heavily on the label quality of each distributed client. In this paper, we consider a federated learning setting with heterogeneous label noise, where each local client might observe training labels with heterogeneous noise rates, which may even be drawn from different subsets of the label space. The above high heterogeneity poses challenges for applying the existing label noise learning approaches to each client locally. We formalize the study of federated learning with heterogeneous label noise by firstly identifying two promising label noise generation models. Then we propose a dual structure approach named FedDual. Intuitively, if there exists a model that filters out the wrongly labeled instances from the local dataset, the effect of label noise can be mitigated. Considering the heterogeneity of local datasets, in addition to the globally shared model, each client in FedDual maintains a local and personalized denoising model. The personalized denoising models can combine information from the global model or other pre-trained models to ensure the performance of denoising. Under this framework, we instantiate our approach with several local sample cleaning methods. We present substantial experiments on MNIST, CIFAR10, and CIFAR100 to demonstrate that FedDual can effectively recognize heterogeneous label noise in different clients and improve the performance of the aggregated model.

1 INTRODUCTION

Federated Learning (FL) aims to learn a common model from different clients while maintaining client data privacy and it has gradually been applied to real applications (Li et al., 2020b; He et al., 2020; 2019). However, data in each local client may be biased (Li et al., 2019b; 2021c; Zhu et al., 2021c) and noisy (Wang et al., 2022). For example, the label quality of each client may be heterogeneous due to human labeling errors (Wei et al., 2022; Han et al., 2018; Yi et al., 2022). The existence of noisy labels severely degrades the generalization performance of FL models (Wang et al., 2022).

We compare the prediction accuracy of FL model training at different noisy label rates in Independent and Identically Distributed (IID) and non-Independent and Identically Distributed (non-IID) distributions in Figure 1(a). The result demonstrates that the performance of the FL model will dramatically decrease with higher label noise, compared with training on clean samples (purple lines). Additionally, the negatice effect of label noise is more severe in the non-IID distribution.

To further diagnose the performance drop observed above, we record the loss of clean samples and corrupted samples during the training process of FL, as shown in Figure 1(b). According to the memorization effect of deep learning when the neural network will prioritize the learning of simple and clean patterns (Liu et al., 2020; Arpit et al., 2017), it is expected that the loss of noisy samples is always larger than the loss of clean samples and it is closer to each other under larger noise rates. The memorization effect is observed in FL models when the data distribution is IID (first row in 1(b)). However, with an increasing degree of non-IID data distribution, the memorization effect is gradually violated: the model started to memorize the corrupted ones equally early or even earlier during the training (bottom right corner on 1(b)). In other words, the curves of noisy samples and clean samples will be switched in high-noise and large-heterogeneity settings, as marked with the red background box of Figure 1. We conjecture that this observation is related to the heterogeneity of label noise in clients, which will be carefully defined in Section 3.



(a)

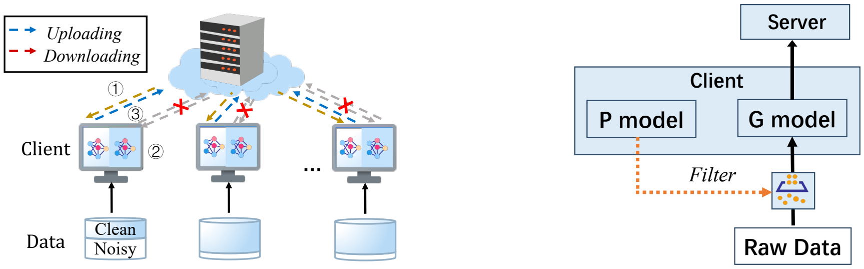
(b)

Figure 1: FedAvg on CIFAR10 with symmetric label noise. (a): Test accuracy when noise rates are 0.0, 0.2, and 0.5. (b): Loss of clean samples and noisy label samples with symmetric label noise. Subfigures in same column are at the same noise level and the noise rate ranges from 0.2 to 0.8 with an interval of 0.3. Subfigures in the same row are at the same heterogeneity. Data heterogeneity is increasing from #class=10 (IID) to #class=2 (extermely non-IID). The case where the clean loss is larger than the noise loss is marked by the red background box.

Although there are many research results on model-based noisy label learning that achieve significant performance by reducing noisy label samples (Cheng et al., 2021a; Li et al., 2019a; Liu & Guo, 2020; Natarajan et al., 2013; Wei et al., 2020), they are often studied in centralized learning and cannot be directly applied to FL because directly applying existing approaches to each client will lead to uncertain performance. The main reason is that with insufficient sample size and possibly an incomplete observed label space (in the non-IID setting) at each local client, the centralized learning-based methods tend to overfit the corrupted samples and recognize the noisy samples as clean samples. Therefore, it is challenging to avoid model overfitting and effectively recognize wrongly labeled samples from different clients in FL.

There are also some studies to solve the challenge of FL with noisy labels and they can be divided into two major types. The first type of methods (Chen et al., 2020; Li et al., 2021a; Yang et al., 2021) solves the challenge by selecting recognized low-noise level clients. However, these methods have limitations. On one hand, unselected clients with clean-label samples are still useful for model training. On the other hand, selected clients with noisy label samples will damage model training to a certain extent. The second type of methods solves the aforementioned limitations to some extent (Tuor et al., 2021; Xu et al., 2022). However, they are often complex with more stages to pre-process or fine-tune global models, and do not fully attend to the heterogeneous label noise setting.

In this paper, we first formalize the definition of homogeneous and heterogeneous label noise, as well as propose two promising label noise generation methods and discuss the situation of homogeneous and heterogeneous label noise in them. Then, we propose an FL framework with a dual model structure named FedDual which can effectively deal with noisy label challenges on FL for both the IID and non-IID distributions. Different from existing works, FedDual is a simple and effective dual model structure to deal with the homogeneous or heterogeneous noise label challenges in the federated setting. Instead of introducing additional complex stages to filter label noise samples, FedDual filters the corrupted label samples in the training process; see Figure 2 for an overview. There are two models in the FedDual, one of the models is the global model (G model) and another model is the personal denoising model (P model) used to filter the noise label samples in clients. The constructed P model filters label noise samples, then the G model will be updated individually based on clean samples in every client. We propose three ways to construct the P model, shown as in Figure 3. The first way is training-free by extracting from pre-trained models trained on other clean datasets, which can acquire a good representation of other clean samples, and filter the noise samples. The second way is extracted from the G model, which can acquire more global knowledge to guarantee the denoising performance of the P model. Both the P model and G model of the third way are trained locally on the clean samples filtered by each other so that different types of error introduced by noisy labels can be filtered with two different learning abilities of models in this exchange filtered samples procedure. Finally, to verify the effectiveness of FedDual, we instantiate FedDual with CORES (Cheng et al., 2021a) and KNN-based denoising (Bahri et al., 2020; Zhu et al., 2021a) methods. Our contributions can be summarized as:



(a) Training process of FedDual.

(b) Dual model structure.

Figure 2: Overview of FedDual. Figure (a) shows the training process of FedDual. In every round, the updated G model from clients will participate in the aggregation process on the server, but the P model will be kept locally. Figure (b) is an illustration of a dual model structure in a client. The P model, as a noisy label samples filter module, helps the G model to select clean samples, based on which the G model can update parameters. When finishing all the local updates, the G model will be sent to the server to be aggregated.

- We discuss and define the problem of federated learning with heterogeneous label noise, as well as propose two label noise generation models for our setting.
- To effectively train the FL model on noisily labeled samples, we propose a dual structure called FedDual, where in addition to the globally shared model, each client maintains a local personalized model to perform the task of denoising samples. The denoising module is flexible to be implemented with many existing label error detection methods.
- With extensive experiments, we demonstrate that FedDual can achieve more significant performance than these baseline methods.

2 Related Work

Robust federated learning Robust federated learning has been studied extensively for statistical and security purposes (Ghosh et al., 2019; Sattler et al., 2019; Ang et al., 2020; Pillutla et al., 2019; Fang et al., 2020; Li et al., 2021b; Wan & Chen, 2021; Xu & Lyu, 2020). Depending on whether the client is malicious or not, research on robust federated learning falls into two main types: 1. robustness in security. 2. robustness in statistics. In research on robustness in security, these methods are often client-level, i.e. malicious clients are viewed as the major factors for weak robustness. On the one hand, many approaches aim to improve the robustness of federated learning by identifying and discarding malicious clients (Xu & Lyu, 2020; Li et al., 2020a). On the other hand, robust aggregation methods are proposed to reweight updates of clients to avoid global model damage by malicious model updates (Fu et al., 2019; Wan & Chen, 2021; Pillutla et al., 2019). There are also some honest clients discarded because of their corrupted samples in local data. Therefore, a large number of important data will be lost if these honest clients are removed. In research on robustness in statistics, except for client-level methods (Fang & Ye, 2022; Yang et al., 2021), some sample-level methods reduce noisy label samples from local data.

Learning with noisy labels There is a large number of works focusing on noisy label learning. They can be mainly divided into five categories: Robust Architecture (Goldberger & Ben-Reuven, 2016; Lee et al., 2019), Robust Regularization (Xia et al., 2020; Gudovskiy et al., 2021), Robust Loss Design (Ghosh et al., 2017; Jiang et al., 2021), and Sample Selection (Jiang et al., 2018; Han et al., 2018; Cheng et al., 2021a). Our work is focused on Sample Selection methods to achieve federated learning with noisy labels. This type of method is gaining ground after that the memorization nature of DNNs has been explored theoretically and empirically to identify clean examples from noisy training data (Xia et al., 2020; Liu et al., 2020).

3 FL with Heterogeneous Label Noise

3.1 Problem Definition

Consider a large collection of data D with a sample size of N and L class labels, which is distributed over K clients that $D = \cup_{k=1}^{K} D^k := \cup_{k=1}^{K} \left\{(x_n^k, y_n^k)\right\}_{n\in[N_k]}$, where $[N_k] = \{1, 2, ..., N_k\}$ is the set of example indices on client k. Each local instance $(x_n^k, y_n^k) \in D^k$ follows the data distribution $(X^k, Y^k) \sim \mathcal{D}^k$. If the distributions $(X^k, Y^k) \sim \mathcal{D}^k, \forall k \in [K]$ are independent and identical, we call the sample from all clients are Independent and Identically Distributed (IID); otherwise they are non-IID. We partition non-IID distribution like (McMahan et al., 2017), where the label space on client k is a subset of the total label space $[L]$. For a machine learning task, the loss of the prediction on examples (x, y) made with model parameters $\boldsymbol{w}$ can be defined as $\ell(x, y; \boldsymbol{w})$. In real-world scenarios where the labels come from human annotators (Wei et al., 2022), the label information is imperfect and the noisy label $\tilde{y}_n$ may or may not be identical to y_n. If $\tilde{y}_n \neq y_n$, we call $\tilde{y}_n$ is corrupted otherwise it is clean. Assume the label noise is class-dependent (Natarajan et al., 2013; Liu & Tao, 2015; Liu & Guo, 2020). Then for each client k, we can use the noise transition matrix T^k to capture the transition probability from clean label $Y = y$ to noise label $\widetilde{Y} = \tilde{y}$. Specifically, we have the following definition:

Definition 1 (Client-Dependent Label Noise)**.** *The label noise on clients $\{1, \cdots, K\}$ is client-dependent if the label noise on each client k can be characterized by label noise transition matrix T^k, where each element satisfies:*

$$\forall k \in [K], i, j \in [L], T_{ij}^k := \mathbb{P}_{(X,Y)\sim\mathcal{D}^k}(\widetilde{Y} = j | Y = i).$$

When $T_k = T, \forall k \in [K]$, we call them homogeneous, otherwise they are heterogeneous. Next, we discuss the generation models for homogeneous and heterogeneous noise, respectively.

3.2 Synthesizing Noisy Labels in FL

Label noise generation Let $p(X) := [\mathbb{P}(Y = 1|X), \cdots, \mathbb{P}(Y = L|X)]^\top$ be the *soft* label of each clean instance (Zhu et al., 2021b). Note in extreme cases when there exists i such that $\mathbb{P}(Y = i|X) = 1$, $p(X)$ reduces to the one-hot encoding of label i. Denote by g the *label noise generation function* specified by noise transition matrix T. According to transition matrix T, noisy labels will be generated, where $\tilde{p}(X) := g(p(X)) = [\mathbb{P}(\widetilde{Y} = 1|X), \cdots, \mathbb{P}(\widetilde{Y} = L|X)]^\top = T^\top p(X)$. Note $Y \sim p(X)$ and $\widetilde{Y} \sim \tilde{p}(X)$. Denote the local noise transition matrix on the client k by T^k. Similarly, we have $Y^k \sim p(X^k)$ and $\widetilde{Y}^k \sim \widetilde{p}(X^k) = (T^k)^\top p(X^k)$.

Data partition Let $M_{K\times L}$ be the data distribution matrix, where M_{ij} is the proportion of the samples labeled as j on the i-th client to the j-th class samples. Denote by h the *data partition mechanism* specified by data distribution matrix M. According to the data distribution matrix M, data D with a sample size of N can be divided into K clients, where $p(X^k) = h(X, p(X), k)$. Further details can be found in the supplementary material; see Appendix B.

Noise generation models in FL Consider two label noise generation models in FL: ANDC (Add Noise then Divide Clients) and DCAN (Divide Clients then Add Noise).

- ANDC: The noisy label is generated by the global noise transition matrix T, where $\widetilde{Y} \sim \tilde{p}(X) = T^\top p(X)$. The corresponding noisy soft label on client k is $\tilde{p}(X^k) := [\mathbb{P}(\widetilde{Y}^k = 1|X^k), \cdots, \mathbb{P}(\widetilde{Y}^k = L|X^k)]^\top = h(X, T^\top p(X), k)$
- DCAN: The noisy label on client k generated by local noise transition matrix T^k satisfies $\widetilde{Y}^k \sim \widetilde{p}(X^k) = (T^k)^\top p(X^k)$ and the corresponding noisy soft label is $\widetilde{p}(X^k) := (T^k)^\top h(X, p(X), k)$.

Next, we discuss when the above label noise generation models are homogeneous or heterogeneous.

Homogeneous Label Noise The partition function h should be the IID partition, i.e., $h(X, p(X), k) = p(X)$. There are two cases:

Algorithm 1 *FedDual*

```
Input: G model $\boldsymbol{w}, \boldsymbol{\theta}$, learning rate $\eta$, number of local epochs $E$, local mini-batch size $B$.
Output: global model $\boldsymbol{w}$.
Server executes:.
    initialize $\boldsymbol{w}_0$;
    for each round $t$ from 1 to T do
        $m \leftarrow \max(C \cdot K, 1)$
        $S_t \leftarrow$ (random set of $m$ clients)                          # selects $m$ clients randomly
        for each client $k \in S_t$ in parallel do
            $\bar{N}_k, \boldsymbol{w}_{t+1}^k \leftarrow$ ClientUpdate($k, \boldsymbol{w}_t$)     # get #clean samples, and the updated G model
        $\boldsymbol{w}_{t+1} \leftarrow \sum_{k=1}^{K} \frac{\bar{N}_k}{\bar{N}} \boldsymbol{w}_{t+1}^k$     # aggregate the updates of G model with FedAvg
Client executes: // Run on client $k$
    ClientUpdate($k$, $\boldsymbol{w}$):
        $\tilde{\mathcal{B}} \leftarrow$ (split $\tilde{D}_k$ into batches of size $B$)
        for each local epoch $i$ from 1 to $E$ do
            for batch $\tilde{b} \in \tilde{\mathcal{B}}$ do
                Obtain $\bar{b}$ = PmodelFilter($\tilde{b}$)     # P model filters out noisy samples for the G model
                $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \bigtriangledown l(\boldsymbol{w}; \bar{b})$     # update the G model
        record the number of selected samples on client $k$ as $\bar{N}_k$.
        Return $\bar{N}_k$, $\boldsymbol{w}$
```

- ANDC (IID partition). With IID partition, the label noise is homogeneous since:

$$\tilde{p}(X^k) := h(X, g(p(X)), k) = h(X, T^\top p(X), k) = T^\top p(X), \forall k \in [K].$$

- DCAN (Homogeneous T). With $g(p(X^k)) = T^\top p(X^k), \forall k \in [K]$ and IID partition, the label noise is homogeneous since:

$$\tilde{p}(X^k) := g(h(X, p(X), k)) = g(p(X^k)) = T^\top p(X^k), \forall k \in [K].$$

Heterogeneous Label Noise In real-world cases, the data partition is often non-IID, i.e., $h(X, p(X), k) = p(X^k) \neq p(X)$. There are two cases:

- ANDC (non-IID partition). With non-IID partition, the label noise is heterogeneous since:

$$\tilde{p}(X^k) := h(X, g(p(X)), k) = h(X, T^\top p(X), k) \neq T^\top p(X^k), \forall k \in [K].$$

- DCAN (Heterogeneous T_k). With $g(p(X^k)) = T^k p(X^k) \neq T^\top p(X^k), \forall k \in [K]$, the label noise is heterogeneous since:

$$\tilde{p}(X^k) := g(h(X, p(X), k)) = g(p(X^k)) = (T^k)^\top p(X^k) \neq T^\top p(X^k), \forall k \in [K].$$

4 FedDual: A Dual Structure Approach

In this section, we introduce FedDual, the proposed dual model structure to effectively train federated learning with noisy labels by identifying and filtering noisy label samples in clients. As a Plug-and-Play component, FedDual can be implemented on the different training strategies of FL. Figure 2 is the overview of FedDual, where the training process of FedDual is exemplified in Figure 2 (a), and the dual model structure is displayed in Figure 2 (b).

4.1 Training process of FedDual

Different from FedAvg (McMahan et al., 2017), the distributed objective of FedDual is:

$$\min_{\boldsymbol{w}} \left\{ f(\boldsymbol{w}) \triangleq \sum_{k=1}^{K} \frac{\bar{N}_k}{\bar{N}} F_k(\boldsymbol{w}) \right\} \tag{1}$$

where $\boldsymbol{w}$ is the parameters of the global model (G model), $\frac{\bar{N}_k}{\bar{N}}$ is the weight of the k-th device, $\frac{\bar{N}_k}{\bar{N}} \geq 0$, and $\sum_k \bar{N}_k = \bar{N}$. Here, $\bar{N}_k$ is the number of selected samples on k-th client and $\bar{N}$ is the

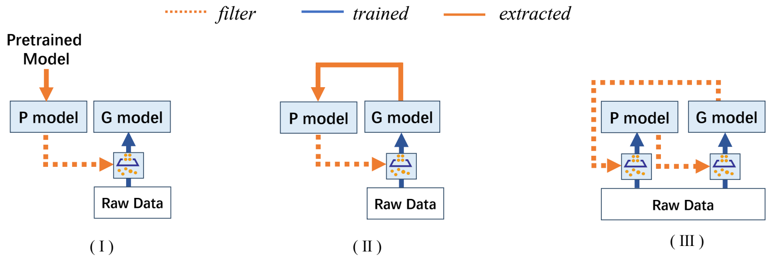


Figure 3: Three types of the dual model structure, corresponding to three ways of obtaining P model.

sum of selected samples on all clients, which depends on the private model (P model) as shown in Figure 2. For client k, F_k is the local optimization objective that the loss of the prediction on local data with noisy labels, which is defined as:

$$F_k(\boldsymbol{w}) = \frac{1}{\bar{N}_k} \sum_{n\in[N_k]} v_n \cdot \ell(x_n, \tilde{y}_n; \boldsymbol{w}),$$

where $v_n \in \{0, 1\}$ indicates whether example n is clean ($v_n = 1$) or not ($v_n = 0$).

Towards the optimization objective, there are three stages included in the training process of Fed-Dualas summarized in Algorithm 1. Particularly, at each communication round t:

- Step-1 (Lines 4–6): The selected clients download the G model from the server. The amount of selected clients is controlled by C, the fraction of a fixed set of K clients in each round.
- Step-2 (Lines 10–17): The lient updates E epochs locally in parallel with the selected clean samples.
- Step-3 (Lines 7–8): The locally-computed parameter updates $\nabla \boldsymbol{w}_t$ of the G model will be uploaded to the server for aggregation.

4.2 Dual model structure

As shown in Figure 3, there are three dual model structures based on the ways to the construct P model. The first way is extracted from pred-trained models trained on other clean datasets. The second way is extracted from the G model of FL. The third way is to train a P model based on the local dataset. Next, we will explain the above three ways in detail.

Training-free It is training-free to construct the P model in the first Dual model structure, as shown in Figure 3 (I). By extracting from pre-trained models trained on other clean datasets, the P model can acquire an effective representation of clean samples. The P model builds on the assumption that nearby representations should belong to the same true class with a high probability (Gao et al., 2016). The representation of feature x_n is denoted by $\hat{x}_n = g(x_n)$, where $g(\cdot)$ denotes a representation extractor and $\hat{x}_n$ is a high-dimensional vector. Based on the k-nearest representations of $\hat{x}_n$, P model then estimates the probability by counting the frequency of each class and get k-NN labels $\hat{y}_n$ (Zhu et al., 2021a). The ith element $\hat{y}_n[i]$ is the probability of predicting class-i and the largest element in $\hat{y}_n$ will be predicted class, i.e. $y_n^{\text{vote}} = \arg\max_{i\in[K]} \hat{y}_n[i]$. We instantiate the KNN-based method by this dual model structure as *FedKNNPretrain*.

Extracted From the G model Figure 3 (II) shows the second dual model structure. In the dual model structure II, the P model is extracted from the G model. According to different de-noising methods, different parts of G model are extracted. For example, CORES (Cheng et al., 2021a) needs to extract the logistic layer of G model to evaluate examples by the regularized loss $\ell(f(x_n), \tilde{y}_n) + \ell_{\mathrm{CR}}(f(x_n))$ and α_n is used to specify thresholds to filter noisy samples, where $\alpha_n := \frac{1}{K} \sum_{\tilde{y}\in[K]} \ell\left(\bar{f}(x_n), \tilde{y}\right) + \ell_{\mathrm{CR}}\left(\bar{f}(x_n)\right)$. Different from CORES, the KNN-based method needs a representation layer to filter noisy samples. In this dual model structure, we respectively instantiate the KNN-based method and CORES as *FedKNN* and *FedCORES*.

Trained Locally In the third of the dual model structure, the P model is trained on local data filtered by the G model. In turn, the G model is trained on the local data filtered by the P model. P model and G model like a twin help each other to filter noisy label samples. Figure 3 (III) is an illustration of how the P model and G model work. There are two key points for the intuition of the third dual network structure. On one hand, the P model can include more individual noisy label information of clients without aggregation but can not accurately identify clean samples because of the lack of global data information. On the other hand, with aggregated data information of different clients, the G model can learn more global representations but lacks client-side personalized noisy label distribution information. Therefore, the dual model structure combines the advantages of the G model and P model to filter noisy label instances. Here, we respectively instantiate the KNN-based method and CORES as *FedTwinKNN* and *FedTwinCORES* in the third dual model structure.

5 EXPERIMENT

5.1 EXPERIMENTS SETUP

Our methods & Baselines. We instantiate two state-of-the-art methods of noisy label learning methods: CORES (Cheng et al., 2021a), and the KNN-based method (Zhu et al., 2021a) to FedDual with three dual model structures. CORES can't be instantiated by dual model structure (I), Because it needs to extract the logistic layer of models. Therefore, there are five corresponding instantiated methods of FedDual as shown below.

- FedKNNpretrain: instantiated KNN-based method according to Dual model structure I.
- FedKNN: instantiated KNN-based method according to Dual model structure II.
- FedCORES: instantiated CORES according to Dual model structure II.
- FedTwinKNN: instantiated KNN-based method according to Dual model structure III.
- FedTwinCORES: instantiated CORES according to Dual model structure III.

We consider CORES (Cheng et al., 2021a), and the KNN-based method (Zhu et al., 2021a) applied to every client (PCORES, PKNN as mentioned below) as baselines. For reference, we also compare FedDual with two state-of-the-art methods of FL: FedAvg (McMahan et al., 2017), and FedProx (Li et al., 2020b), and FL based on Loss Correction (Patrini et al., 2017) The detailed explanation is as follows:

- PCORES: CORES applied to each client locally to sieve noisy label samples.
- PKNN: KNN-based method applied to each client locally to sieve noisy label samples.
- FedAvg: the vanilla FL framework.
- FedProx: A popular federated learning optimizer that adds a quadratic penalty term to the local objective.
- FedCorAvg: FL based on Loss Correction.

Implement Details. We evaluate different methods on MNIST,CIFAR10, CIFAR100 datasets. We consider two noisy label data generation models and corrupt these datasets with two types of label noise: symmetric and pairflip (Cheng et al., 2021a). We generated heterogeneous label noise by assigning each of the clients 5 classes for 10 classification tasks and 50 classes for 100 classification tasks. In addition, we set common parameter for FL as: local epoch E = 5, the fraction of selected clients on CIFAR100 and MNIST per round C = 0.1, Local solver is SGD, batch size B = 32, learning rate η = 0.01. Further details can be found in the supplementary material; see Appendix C.

We use test accuracy to evaluate the performance of the federated learning task and F_1-score to evaluate the performance of label error detection. Particularly, F_1-score is calculated by $F_1 = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$. Let $v_n = 1$ indicate that $\tilde{y}_n$ is detected as a corrupted label, and $v_n = 0$ if $\tilde{y}_n$ is detected to be clean (Cheng et al., 2021b). The precision and recall can be calculated as $\text{Precision} = \frac{\sum_{n\in[N]} \mathbb{1}(v_n=0,\tilde{y}_n=y_n)}{\sum_{n\in[N]} \mathbb{1}(v_n=0)}$, $\text{Recall} = \frac{\sum_{n\in[N]} \mathbb{1}(v_n=0,\tilde{y}_n=y_n)}{\sum_{n\in[N]} \mathbb{1}(\tilde{y}_n=y_n)}$.

5.2 COMPARISON WITH STATE-OF-THE-ART METHODS

We compare the prediction performance of FedDual with all the baseline methods as mentioned above, in homogeneous and heterogeneous label noise settings with different types and noise rates.

Table 1: The accuracies of various methods on CIFAR10 with homogeneous label noise and heterogeneous label noise at different noise levels. Both two label noise generation models (ANDC and DCAN) are tested. The accuracy where FedDual is better than the existing noise label methods applied to FL is painted with a darker background color. For each noise level, we highlight in bold all cases when the accuracy of FedDual is better than other baselines.

	Method	Homogeneous label noise					Heterogeneous label noise				
			Symmetric		Pairflip			Symmetric		Pairflip	
		0.0	0.2	0.5	0.2	0.4	0.0	0.2	0.5	0.2	0.4
ANDC	FedAvg	89.00	80.03	54.85	81.75	59.81	83.23	71.19	45.42	63.08	48.82
	FedCorAvg	77.51	52.08	58.65	71.22	58.92	69.75	57.24	51.86	58.27	41.27
	FedProx	81.39	73.53	60.35	75.67	61.02	75.83	68.47	54.66	68.02	50.02
	PCORES	52.13	44.21	33.14	37.88	36.21	51.95	41.10	27.70	36.73	34.76
	PKNN	87.71	79.48	57.06	81.03	59.82	82.84	72.76	48.64	65.71	48.22
	FedCORES	**89.52**	**86.15**	54.59	**87.40**	**69.15**	**85.17**	**81.14**	**59.35**	67.47	**51.23**
	FedTwinCORES	**90.27**	**84.74**	**65.53**	**81.89**	**69.07**	**86.83**	**78.54**	**57.47**	57.14	46.42
	FedKNN	86.96	**84.28**	**70.67**	**83.85**	**70.59**	82.53	**78.52**	**63.40**	66.63	49.24
	FedTwinKNN	88.14	**82.30**	59.63	**83.45**	**63.22**	**84.37**	**75.37**	52.18	66.73	47.72
	FedKNNPretrain	86.53	**84.07**	**72.73**	**83.37**	**65.74**	83.17	**78.86**	**64.30**	**68.69**	47.95
DCAN	FedAvg	89.00	78.19	52.46	80.21	58.54	83.23	61.37	31.55	69.43	54.09
	FedCorAvg	77.51	59.80	56.86	71.31	55.03	69.75	55.65	39.39	58.57	45.41
	FedProx	81.39	73.49	60.58	75.65	61.34	75.83	63.02	43.90	65.41	54.75
	PCORES	52.13	43.81	30.58	38.28	36.73	51.95	35.17	23.42	37.51	33.36
	PKNN	87.71	80.63	57.14	79.90	55.39	82.84	68.3	35.68	68.72	49.67
	FedCORES	**89.52**	**86.17**	56.91	76.21	**68.75**	**85.17**	**77.54**	33.90	**70.69**	**60.83**
	FedTwinCORES	**90.27**	**84.99**	**65.18**	**80.60**	**65.38**	**86.83**	**73.95**	43.22	**75.16**	**62.04**
	FedKNN	86.96	**84.81**	**74.01**	**84.80**	**67.34**	82.53	**79.05**	**52.88**	**77.36**	53.88
	FedTwinKNN	88.14	**83.36**	59.69	**82.22**	57.38	**84.37**	**72.73**	35.78	**70.77**	50.02
	FedKNNPretrain	86.53	**84.80**	**74.91**	**83.73**	**65.66**	83.17	**81.20**	**60.31**	**78.55**	54.48

Table 2: The accuracies of various methods on CIFAR100 with heterogeneous label noise (DCAN) at different noise levels.

Method	CIFAR100					MNIST				
	Clean	Symmetric		Pairflip		Clean	Symmetric		Pairflip	
	0.0	0.2	0.5	0.2	0.4	0.0	0.2	0.5	0.2	0.4
FedAvg	66.64	47.00	22.57	47.69	37.56	98.48	91.72	75.93	96.39	91.33
FedCorAvg	28.08	24.12	17.25	23.7	18.92	82.58	81.24	37.85	92.68	56.87
FedProx	63.09	48.51	24.2	50.51	39.0	98.03	92.38	85.03	96.10	92.14
FedPCORES	40.17	29.52	14.77	31.37	24.4	62.98	56.58	46.89	57.68	55.17
FedPKNN	64.66	52.31	26.16	52.23	36.89	98.07	94.78	84.34	95.62	88.12
FedCORES	**66.89**	**54.43**	20.48	**53.16**	35.66	98.74	**95.93**	66.84	96.29	93.37
FedTwinCORES	66.00	**49.88**	21.95	**53.73**	**40.08**	98.84	93.49	68.81	**96.35**	92.47
FedKNN	**66.75**	**61.27**	**31.71**	**58.71**	**44.25**	98.07	**97.72**	**86.57**	**97.52**	92.29
FedTwinKNN	66.49	**56.03**	25.94	**55.15**	**39.94**	98.12	**97.79**	**88.85**	**97.49**	**90.44**
FedKNNpretrain	66.49	49.85	22.46	**52.79**	38.84	98.70	**95.18**	**88.13**	96.48	88.81

The results can be found in Table 1 for the CIFAR10 dataset, Table 2 for the CIFAR100 dataset and the MNIST dataset. More results can be found in the supplementary. On the one hand, we can find that FedDual can achieve the best accuracy whatever with homogeneous or heterogeneous label noise at the different noise type and noise level, compared to all the baselines. On the other hand, we find that PCORES and PKNN are almost ineffective when applied directly to the local client to filter noisy label samples, where PCORES is even outright collapse. Take into consideration the compatibility, we also test the performance of of FedDual on datasets without noisy label samples. The experimental results show that FedDual achieves approximated prediction accuracy to FedAvg and FedProx when noisy label rate is 0.0.

By analyzing these baselines and five instantiations of dual model structure, we also find interesting results. Firstly, the performance of PCORES is worse seriously than PKNN. The underlying cause of this phenomenon is that CORES is a model training-based method to filter noisy label samples. When applied to local clients, CORES are too easy to overfit the noise samples because of the insufficient sample size. Different with CORES, although supervised by noisy label samples, PKNN detector only extract model representation, which can greatly avoid overfitting to the noise samples. However, PKNN doesn't effectively filter noisy label samples because of the invalid representation.

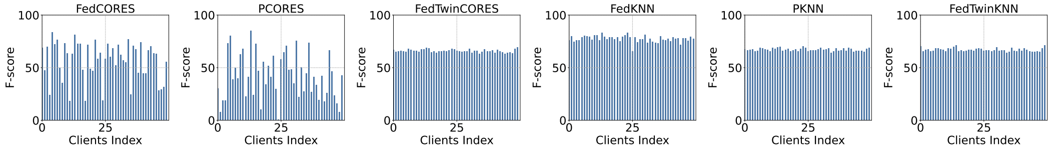


Figure 4: The F-score distribution of FedDual in different clients with heterogeneous label noise (class=5) and noise rate 0.5. The fraction of selected clients is $C = 1$.

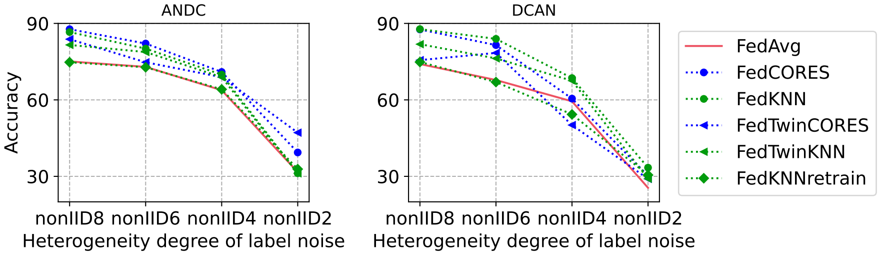


Figure 5: The accuracies of FedDual with increasing heterogeneity of label noise on the CIFAR10 dataset with noise rate 0.2.

Secondly, the performance of FedDual with dual structure III is not highly related with the performance of base learner. Take CORES as an example, even though trained on local data is completely collapsed (refer to PCORES), P model performance is greatly improved in dual structure III (refer to FedTwinCORES). Take KNN as a counterexample, even though trained on local data (refer to PKNN) performs well, P model performance in FedTwinKNN is worse than FedTwinCORES.

5.3 Denoising stability of FedDual

Figure 4 shows the F-score distribution of FedDual in different clients to test the denoising stability of FedDual. By comparing the three dual structures, we can conclude that all the instantiation method of structure (III) ensures denoising stability in each client. By comparing the instantiations of FedDual, KNN-based methods whatever PKNN, FedKNN, or FedTwinKNN, achieve client noisy labels identification with less gap. Note that FedKNN achieves higher noisy label identification than PKNN, this is because of the good representation of the P model extracted from the G model, which is consistent with the conclusion in Section 5.2. For CORES based method, although FedCORES can achieve higher noisy label identification than PCORES generally, the noisy label identification of FedCORES is not stable. As an alternative solution, FedTwinCORES can achieve more stable noisy label identification on different clients.

5.4 Robustness of FedDual with Heterogeneous Label Noise

Figure 5 shows how FedDual's performance with the increasing increasing heterogeneity of label noise. In general, the performance of FedDual will decrease with the increasing heterogeneity of label noise. However, all the instantiations of FedDual almost achieve higher accuracy, compared with FedAvg. Among these methods, FedKNNpretrain achieve the lowest accuracy in different degree of heterogeneous label noise.

6 Conclusions

In this paper, we discuss federated learning with heterogeneous label noise by formalizing heterogeneous label noise under federated learning and proposing a simple and effective dual model structure to filter noisy label samples called FedDual to solve the challenge. The extensive experiments demonstrate the outperformance of FedDual at homogeneous and heterogeneous label noise with different noise rates.

References

Fan Ang, Li Chen, Nan Zhao, Yunfei Chen, Weidong Wang, and F Richard Yu. Robust federated learning with noisy communication. *IEEE Transactions on Communications*, 68(6):3452–3464, 2020.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pp. 233–242. PMLR, 2017.

Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pp. 540–550. PMLR, 2020.

Yiqiang Chen, Xiaodong Yang, Xin Qin, Han Yu, Biao Chen, and Zhiqi Shen. Focus: Dealing with label quality disparity in federated learning. *arXiv preprint arXiv:2001.11359*, 2020.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021a. URL `https://openreview.net/forum?id=2VXyy9mIyU3`.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021b.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

Xiuwen Fang and Mang Ye. 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.

Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. *arXiv preprint arXiv:1912.11464*, 2019.

Wei Gao, Xin-Yi Niu, and Zhi-Hua Zhou. On the consistency of exact and approximate nearest neighbor with noisy data. *CoRR*, abs/1607.07526, 2016. URL `http://arxiv.org/abs/1607.07526`.

Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

Denis Gudovskiy, Luca Rigazio, Shun Ishizaka, Kazuki Kozuka, and Sotaro Tsukizawa. Autodo: Robust autoaugment for biased data with label noise via scalable probabilistic implicit differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16601–16610, 2021.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, and Ji Liu. Central server free federated learning over single-sided trust social networks. *arXiv preprint arXiv:1910.04956*, 2019.

Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. An information fusion approach to learning with instance-dependent label noise. In *International Conference on Learning Representations*, 2021.

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pp. 3763–3772. PMLR, 2019.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2019a.

Li Li, Huazhu Fu, Bo Han, Cheng-Zhong Xu, and Ling Shao. Federated noisy client learning. *arXiv preprint arXiv:2106.13239*, 2021a.

Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020b.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations, ICLR*, 2019b.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fed{bn}: Federated learning on non-{iid} features via local batch normalization. In *International Conference on Learning Representations*, 2021c. URL `https://openreview.net/pdf?id=6YEQUn0QICG`.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pp. 6226–6236. PMLR, 2020.

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

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 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.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Felix Sattler, Simon Wiedemann, Klaus-Robert Muller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. Overcoming noisy and irrelevant data in federated learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5020–5027. IEEE, 2021.

Ching Pui Wan and Qifeng Chen. Robust federated learning with attack-adaptive aggregation. *arXiv preprint arXiv:2102.05257*, 2021.

Zhuowei Wang, Tianyi Zhou, Guodong Long, Bo Han, and Jing Jiang. Fednoil: A simple two-level sampling method for federated learning with noisy labels. *arXiv preprint arXiv:2205.10110*, 2022.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13726–13735, 2020.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=TBWA6PLJZQm`.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*, 2020.

Jingyi Xu, Zihan Chen, Tony QS Quek, and Kai Fong Ernest Chong. 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.

Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. *arXiv preprint arXiv:2011.10464*, 2020.

Miao Yang, Hua Qian, Ximin Wang, Yong Zhou, and Hongbin Zhu. Client selection for federated learning with label noise. *IEEE Transactions on Vehicular Technology*, 71(2):2193–2197, 2021.

Li Yi, Sheng Liu, Qi She, A Ian McLeod, and Boyu Wang. On learning contrastive representations for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16682–16691, 2022.

Zhaowei Zhu, Zihao Dong, Hao Cheng, and Yang Liu. A good representation detects noisy labels. *arXiv preprint arXiv:2110.06283*, 2021a.

Zhaowei Zhu, Tianyi Luo, and Yang Liu. The rich get richer: Disparate impact of semi-supervised learning. *arXiv preprint arXiv:2110.06282*, 2021b.

Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pp. 3–4, 2021c.