LEARNING CRITICALLY IN FEDERATED LEARNING WITH NOISY AND HETEROGENEOUS CLIENTS

Anonymous authors
Paper under double-blind review

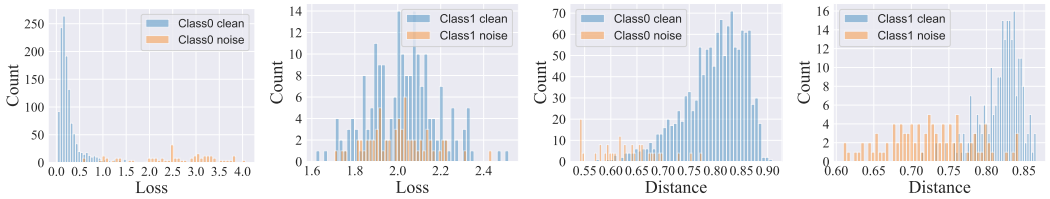
ABSTRACT

Federated learning (FL) is a distributed learning framework for collaboratively training models with a privacy guarantee. Class imbalance problem is the main problem in FL with heterogeneous clients. Besides, Label noise is also an inherent problem in scenarios since clients have varied expertise in annotations. However, *the co-existence of heterogeneous label noise and class-imbalance distribution in FL's small local datasets* renders conventional label-noise learning methods ineffective. Thus, in this paper, we propose algorithm **FEDCNI**, including a noise-resilience local solver and a robust global aggregator, to address the challenges of noisy and highly-skewed data in FL without using an additional clean proxy dataset. For the local solver, we first design a prototypical classifier to detect noisy samples by evaluating the similarity between samples and prototypes. Then, we introduce a curriculum pseudo labelling method with thresholds for different classes cautiously from the noisy samples. For the global aggregator, We aggregate critically by switching re-weighted aggregation from data size to noise level in different learning periods. Experiments on real-world datasets demonstrate that our method can substantially outperform state-of-the-art solutions and is robust in mix-heterogeneous FL environments.

1 INTRODUCTION

The proliferation of smart devices such as mobile phones, cameras, and sensors has dramatically expanded the perception and capabilities of numerous distributed devices, forming an increasingly intelligent Internet of Things (Pandey et al., 2020) network. The massive data plays a key role in generating powerful predictive models to provide better services to users. However, transferring users' data to the server poses a high privacy risk to service providers and mobile users and renders traditional centralized training ineffective. Therefore, Federated Learning (FL) (McMahan et al., 2017; Li et al., 2020; Acar et al., 2021) stands out as a promising solution that enables such collaborative training only by aggregating local models uploaded from clients without any data exchange.

In practical federated learning implementations on real heterogeneous networks, labels of local data are often machine-generated or manually annotated. Nevertheless, clients have different domain expertise and various human biases in annotation, resulting in inaccurate local labels and heterogeneous label noise among clients (Fang & Ye, 2022; Xu et al., 2022; Yang et al., 2022). It is shown that the noisy labels result in overfitting and memorization of the noisy data (Arpit et al., 2017; Zhang et al., 2021b), thus, causing catastrophic failure of deep neural networks (Han et al., 2018; Li et al., 2019). Although there are many works tackling noisy-label learning problems in centralized training, they are not effective in FL due to inherent challenges in FL. The most dominant challenge is the co-existence of label noise and class-imbalance distribution in clients' local small data. As we show in Figure 1, the small-loss technique (Li et al., 2019; Han et al., 2018; Jiang et al., 2018), which is the most commonly used technique in noise detection of centralized learning, no longer works when the client's data has imbalanced class, noisy label, and small size. The small dataset makes the model poorly generalized overall. Besides, the classifier learned directly from class-imbalanced data is biased towards the majority class, which further leads to a poor generalization of minority classes. Label noise cannot be detected using the sample loss produced by such a biased classifier because both clean and noisy samples of the minority class have large losses and low confidence, which is hard to distinguish. Other challenges, like data privacy and different noise



(a) Small-loss on majority (b) Small-loss on minority (c) Our on majority (d) Our on minority

Figure 1: **Illustration of noise detection results on small and class-imbalanced local datasets.** (a) and (b) show that the small-loss method is able to distinguish noisy samples in the majority class but fails in the minority class. However, in (c) and (d), our proposed prototypical method can be used to divide noisy and clean samples into both majority and minority.

levels across clients, are also crucial in noisy-label FL. Thus, effective methods need to be developed in FL with noisy labels to tackle these challenges jointly.

There are some pioneer works addressing the problem of FL with noisy labels. However, the strong assumptions or preconditions prevent them from being effective and general enough in practical situations. Some works rely on clean proxy datasets on the server (Tuor et al., 2021; Chen et al., 2020; Yang et al., 2021), clean public datasets held by all clients (Fang & Ye, 2022), or clean clients without noisy labels (Xu et al., 2022). However, on one hand, clean datasets are not realistic since collecting the clients' data is forbidden in FL and annotating such a clean dataset requires huge costs. On the other hand, the clean client assumption is not satisfied if the FL system is rather heterogeneous. Besides, current methods are not tailored to solve the class-imbalance heterogeneity in noisy-label learning, which makes them ineffective in mix-heterogeneous FL environments.

In this paper, we propose **Fed**erated **C**ritical learning for **N**oisy and **I**mbalanced clients (**FEDCNI**) to address the challenges of label noise and co-occurrence of class imbalance without additional clean datasets in FL. We formulate the noisy-label FL as a bi-level problem that *clients learn from noisy data* and *server learns from noisy clients*. Since the noisy data can cause catastrophic failure in training and when the local dataset is small, the failure is more dominant, FEDCNI learns critically and cautiously from the rare and noisy local data. FEDCNI detects the clients' local noisy samples and measures the noise levels of clients. Then it learns critically from the detected noisy samples and aggregates critically according to the noise levels of clients.

Specifically, in FEDCNI, we devise a noise-resilience local solver and a robust global aggregator. The noise-resilience local solver consists of two parts, namely, prototypical noise detection and noise-resilience local loss. In the prototypical noise detection, we calculate the class prototypes in each local epoch and use cosine similarity to distinguish samples with wrong labels. It is shown that the prototypical technique is more effective than small-loss methods to detect noisy samples in class-imbalanced data (Wei et al., 2021; 2022), and we find it also works in FL's small local datasets (shown in Figure 1). To avoid overfitting on majority classes and noisy data, we introduce a dynamic threshold for pseudo labelling, considering the class's noise level and quantity. Then, the noise-resilience local loss is introduced to treat the clean and noisy samples differently for critical learning. As to the global aggregator, a noise level re-weighted aggregation and data-size weighted aggregation in different learning periods provide a balance between data quality and quantity. Our contributions can be concluded as follows:

- We propose the FEDCNI tailored for tackling the joint challenge of noise label and class-imbalance data in real-world FL scenarios.
- We craft a noise-resilience local solver and a robust global aggregator without offending clients' privacy, extra communication overhead and additional monitor or proxy dataset, performing robustly in detection precision and other metrics.
- We conduct extensive experiments to show that our proposed approach outperforms state-of-the-art FL methods on multiple datasets.

The rest of this paper is organized as follows. We introduce the related work of noisy-label federated learning background in Section 2. The problem formulation is given in Section 3. The main algorithm is presented in Section 4. In Section 5, experiments are conducted to evaluate the performance of our method. Finally, Section 6 concludes this paper.

2 RELATED WORKS

Heterogeneous Data in Federated Learning. Clients have Non-IID data distributions is an inherent problem in FL, which is also known as data heterogeneity. There are mainly two kinds of heterogeneity in FL, one is label distribution shift, and the other is feature distribution shift (Li et al., 2022). Label distribution shift results from a class-imbalanced local dataset for each client, and there are a lot of works in FL that focus on designing effective algorithms to tackle this heterogeneity. Li et al. propose FEDPROX with a proximal term on the client side so the model parameters obtained by the client after local training will not deviate too much from the initial server parameters. FEDDYN (Acar et al., 2021) adds a regularization term in local training based on the global model and the model from previous rounds of communication to overcome device heterogeneity. Karimireddy et al. propose SCAFFOLD to control variables for reducing client drift.

Noisy-label Federated Learning. There are some previous works that have concerned the FL with noisy labels. Xu et al. propose multi-stage federated learning including noise client detection, noisy sample detection and correction, and a vanilla FEDAVG (McMahan et al., 2017) phase. However, their multi-step LID score calculations require a high computational complexity, and there are extensive hyper-parameters that need to be settled. They adopt the small-loss technique, which works poorly in a class-imbalance scene. The method also requires clean clients, which is not realistic in practice. Yang et al. introduce the exchange of class centroids between the server and clients to give pseudo labels and generate loss based on similarity, which may threaten clients' privacy due to the direct transfer of class centroids. They also adopt the small-loss technique. Li et al. present a robust aggregation with data quality and quality measurement. There are also some existing works that rely on a clean proxy (benchmark) dataset on the server side. Tuor et al. upload local samples' loss distribution to the server for noise detection by proxy dataset. The transmitted loss distribution can raise severe privacy concerns. Fang & Ye interchange model logits and analyze them by a public dataset. The method lacks noise detection, only depending on KL divergence of knowledge. Besides, the auxiliary public dataset is not available for the server and clients in real applications.

3 PROBLEM FORMULATION

We consider a typical federated learning scenario with a multi-class classification task. There are K clients and overall N data samples in the training. Each client $k \in \{1, \dots, K\}$ holds a private dataset $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, where x_i^k is the input of the training sample, corresponding y_i^k denotes the given label, and the number of local samples is n_k ($\sum_{k=1}^{K} n_k = N$). In the inaccurate annotation scenario, $y_i^k \in \{1, \dots, C\}$ can be the same as the ground-truth label $\bar{y}_i^k$, or be different as a noise. In FL, the global model θ aims to minimize the sum loss over all clients, formalized as

$$\min_{\theta} \mathcal{L}(\theta) = \sum_{k=1}^{K} \frac{n_k}{N} \mathcal{L}_k(\theta), \tag{1}$$

where $\mathcal{L}_k(\theta)$ is the local loss on $\mathcal{D}_k$ for client k. It is formulated as

$$\mathcal{L}_k(\theta) = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathcal{L}_{ce}\left(y_i^k, \mathbb{P}(\theta; x_i^k)\right), \tag{2}$$

where $\mathcal{L}_{ce}$ is the cross entropy (CE) loss function and $\mathbb{P}(\theta; :)$ is the prediction softmax given model θ. FL algorithms solve the above optimization problem by iterating between the local training of clients (initialized by the global model θ) and the global aggregation of clients' models. Specifically, client k updates the received global model ($\theta_k \leftarrow \theta$) by stochastic gradient descent (SGD) in each iteration, defined as follows:

$$\theta_k \leftarrow \theta_k - \eta \nabla \mathcal{L}_k(\theta_k), \tag{3}$$

where η refers to the local learning rate. The local training is conducted by E epochs. The central server then aggregates the clients' models by:

$$\theta \leftarrow \sum_{k=1}^{K} \frac{n_k}{N} \theta_k. \tag{4}$$

However, due to the existence of noisy labels, the optimization of minimizing loss leads to the model overfitting on noisy labels. Aggregating these biased local models will make the global model with poor generalization performance. Therefore, approaches for reducing the effect of noisy samples are needed to achieve this optimization goal.

4 PROPOSED METHOD

Our method FEDCNI consists of three basic modules, *the prototypical local noise detection*, *the noise-resilience local loss*, and *the robust global aggregator*. The prototypical local noise detection and the noise-resilience local loss form *the noise-resilience local solver*. The overall pseudo-code is shown in Algorithm 1, and we will introduce the three modules respectively.

Algorithm 1: FEDCNI Algorithm

Input: Clients $\{1, \ldots, K\}$, local dataset $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, switching round T_s, global round T.
Output: Gloabl model θ^T.
Initialize global model weights θ^1; **for** $t = 1, \ldots, T$ **do**
  for client $k = 1, \ldots, K$ ***in parallel*** **do**
    $\theta_k^t \leftarrow \theta^t$;
    for local epoch $e = 1, \ldots, E$ **do**
      for class $c = 1, \ldots, C$ **do**
        Calculate prototype $p_{k,c}$ via Eq. (5);
        Calculate similarity $\cos(p_{k,c}, \mathcal{F}(\theta_k; x_i^k))$ between each sample and prototype;
      end
      Divide $\mathcal{D}_k$ into clean subset $\mathcal{C}_k$ and noise subset $\mathcal{N}_k$ by GMM; Update $\tau_{k,c}$ by Eq. (7, 8);
      for $x_i^k \in \mathcal{N}_k$ **do**
        Compute the pseudo-label $\widetilde{y}_i^k \leftarrow \arg\max_c(\{\cos(p_{k,c}, \mathcal{F}(\theta_k; x_i^k))\}_{c=1}^C)$; **if** $t \geq T_s$ and $\cos(p_{k,\widetilde{y}_i^k}, \mathcal{F}(\theta_k; x_i^k)) \geq \tau_{k,c}$ **then**
          $y_i^k \leftarrow \widetilde{y}_i^k$;
        end
      end
      Obtain $\mathcal{L}_k^{sum}$ by Eq. (11,12,13);
      $\theta_k^t \leftarrow \theta_k^t - \eta \nabla \mathcal{L}_k^{sum}(\theta_k^t)$;
    end
    Return $|\mathcal{C}_k|$, $|\mathcal{D}_k|$, θ_k^t;
  end
  The server updates θ^{t+1} by Eq. (14).
end

4.1 PROTOTYPICAL LOCAL NOISE DETECTION

In the local training phase, we initially conduct prototypical local noise detection in each client, and it includes three steps. The first step is to calculate the prototypes of each class. Second, we calculate the cosine similarities between the samples and the corresponding prototypes and use Gaussian Mixture Model on these similarities to detect noisy samples. Third, given the detected noisy samples, we dynamically adjust the confidences of classes and give pseudo labels according to the confidence to prudently learn from the noisy data.

Prototype Generation. The class prototype p_c is defined as a normalized mean of samples' embeddings for the class $c \in \{1, \ldots C\}$. For client k, we can obtain its local prototypes as:

$$p_{k,c} = \frac{1}{|\mathcal{D}_{k,c}|} \sum_{(x_i^k, y_i^k) \in \mathcal{D}_{k,c}} \mathcal{F}(\theta_k; x_i^k), \tag{5}$$

where $\mathcal{D}_{k,c}$ denotes the samples given the label c in local dataset $\mathcal{D}_k$ and $\mathcal{F}(\theta_k; x_i^k)$ refers to the output embedding of sample x_i^k given the model θ_k.

We note that previous works adopt prototype sharing from clients to server (Tan et al., 2022; Yang et al., 2022) and it will violate the data privacy of clients. Whilst we compute the local prototypes just for local noise detection for each client. Our local prototype solution is also effective in noise detection and further safeguards client privacy.

Noise Detection. For client k, given a prototype of a class $p_{k,c}$, we compute the embeddings of the samples which are labelled as c. Intuitively, the embeddings of clean samples may have high similarities to the prototype, while the noisy samples represent outliers in the embedding space. Thus, we use a two-component Gaussian Mixture Model (GMM) in the similarities to distinguish noisy and clean samples for each class. It is worth mentioning that there are two common similarity measurements for the embeddings, based on Euclidean space (i.e. L_2 distance) (Wei et al., 2021)

and inner product space (e.g. cosine function) (Wei et al., 2022). The embeddings are usually high dimensional (512 in our paper) and the inner product measurements are more effective when the vectors have high dimensions. Thus, we adopt the cosine function as the similarity measurement. The cosine similarity between a sample $x_i^k \in \mathcal{D}_k$ and a prototype $p_{k,c}$ is given by:

$$\cos(p_{k,c}, x_i^k) = \frac{\langle p_{k,c}, \mathcal{F}(\theta_k; x_i^k)\rangle}{\|p_{k,c}\|\|\mathcal{F}(\theta_k; x_i^k)\|}. \tag{6}$$

We can obtain class-wise similarity sets $s_{k,c} = \{\cos(p_{k,c}, x_i^k) | x_i^k \in \mathcal{D}_{k,c}\}$ that contain the similarities between the class prototype and the samples. We conduct local noise detection using a two-component GMM in $s_{k,c}$ and obtain the noisy set with lower similarity $\mathcal{N}_{k,c}$ and clean set with higher similarity $\mathcal{C}_{k,c}$. Then the overall detected noisy set for client k is $\mathcal{N}_k = \sum_{c=1}^{C} \mathcal{N}_{k,c}$, and the clean set is $\mathcal{C}_k = \sum_{c=1}^{C} \mathcal{C}_{k,c}$. The noisy and clean sets are disjoint that $\mathcal{N}_k \cap \mathcal{C}_k = \emptyset, \mathcal{N}_k \cup \mathcal{C}_k = \mathcal{D}_k$.

Curriculum Pseudo Labelling. Existing noisy samples may have a detrimental influence on optimization and generalization. To address the noisy-label challenge in the small-size local data, we tackle it from a semi-supervised learning view (Li et al., 2019; Zhang et al., 2021a; Cascante-Bonilla et al., 2021). The detected noisy samples are regarded as unlabeled data, and we assign pseudo labels to these samples. However, the class imbalance in the local data causes different difficulties in labelling majority classes and minority classes. Intuitively, the majority classes are easier to discern and learn, so the labelling confidence is always high. At the same time, due to the rareness, the model has bad memorization of the minority classes. Even though some noisy samples' ground-truth labels belong to a minority class, the labelling confidence is low. Inspired by the idea of curriculum labelling in semi-supervised learning (Zhang et al., 2021a), we propose a novel curriculum pseudo labelling method considering the noise level and data quantity of each class.

We define the dynamic threshold $\tau_{k,c}$ for each class c in client k. Every class $c \in \{1, \ldots, C\}$ has a different level of noise and data size, therefore, $\tau_{k,c}$ is to describe the difficulty of learning this class. First, we introduce the definition of learning difficulty $\rho_{k,c}$ as:

$$\rho_{k,c} = \frac{\sum_{(x_i^k, y_i^k) \in \mathcal{C}_{k,c}} \mathbb{I}(\max_j(\mathbb{P}^j(\theta_k; x_i^k)) > \tau_{k,c}) \cdot \mathbb{I}(\arg\max_j(\mathbb{P}^j(\theta_k; x_i^k)) = c)}{|\mathcal{D}_{k,c}|}, \tag{7}$$

where $\mathbb{P}(\theta_k; x_i^k)$ is the prediction softmax values and $\mathbb{P}^j(\theta_k; x_i^k)$ is the j-th indexed value of the softmax. In Equation 7, the initial threshold $\tau_{k,c}$ is set as τ for all classes after the early phase of training. In the equation, we quantify the learning difficulty $\rho_{k,c}$ as a confident and clean sample proportion for each class. Then, we normalize $\rho_{k,c}$ to update the dynamic threshold $\tau_{k,c}$, as:

$$\tau_{k,c} = \frac{\rho_{k,c}}{\max(\rho_k)}\tau, \text{ where } \rho_k = \{\rho_{k,c} | c \in \{1, \ldots, C\}\}. \tag{8}$$

We update $\rho_{k,c}$ and $\tau_{k,c}$ iteratively as in Equation 7 and Equation 8 during local training.

Besides, we can also get the cosine similarities between one sample x_i^k and all classes' prototypes $\{\cos(p_{k,c}, \mathcal{F}(\theta_k; x_i^k))\}_{c=1}^{C}$, and we use these cosine similarities as the prototypical classifier to predict the pseudo labels for the detected noisy set $\mathcal{N}_k$. Concretely, given a sample $(x_i^k, y_i^k) \in \mathcal{N}_k$, the pseudo label is as

$$\widetilde{y}_i^k = \arg\max_c(\{\cos(p_{k,c}, \mathcal{F}(\theta_k; x_i^k)) | c \in \{1, \ldots, C\}\}). \tag{9}$$

According to the dynamic thresholds, if the cosine value of the pseudo label is higher than the threshold, we assign the sample with the pseudo label, otherwise, we use its original label, as

$$y_i^k = \begin{cases} \widetilde{y}_i^k, & \text{if } \cos(p_{k,\widetilde{y}_i^k}, \mathcal{F}(\theta_k; x_i^k)) > \tau_{k,\widetilde{y}_i^k}, \\ y_i^k, & \text{otherwise.} \end{cases} \tag{10}$$

In practice, we find the accuracy of pseudo labelling is relatively low at the beginning (see Figure 9 of the appendix), so we set a switching round T_{s_1}: before T_{s_1}, we use the given labels for the noisy samples, and after T_{s_1}, we use the pseudo labels as Equation 10. The experiment in Figure 9 of the appendix verifies that this strategy can improve the generalization in the early training.

4.2 NOISE-RESILIENCE LOCAL LOSS

Given the detected noisy samples and the corresponding pseudo label in Section 4.1, we now devise the noise-resilience local loss to critically learn from the noisy data. The loss consists of two parts, the first is the denoise mixup loss and the second is the prototypical similarity loss, and we treat the detected clean and noisy samples differently in each loss. Note that unlike previous works in noisy-label FL Xu et al. (2022); Fang & Ye (2022), we are not using the vanilla CE loss; we empirically find that the simple CE loss will cause performance degradation in FL's small, noisy, and imbalanced local datasets; instead, our loss is robust and noise-resilience (evidence in Table 6 of the appendix).

Denoise Mixup Loss. Recall that Mixup (Zhang et al., 2018) is a data augmentation method that mixes up the samples' features and labels to generate new samples. Specifically, it generates new sample $(\tilde{x}, \tilde{y})$ by linear combination of randomly selected pairs of samples (x_i, y_i) and (x_j, y_j), as $\tilde{x} = \lambda x_i + (1-\lambda)x_j, \tilde{y} = \lambda y_i + (1-\lambda)y_j$. Mixup has been proven to be effective in both semi-supervised learning (Zhang et al., 2021a) and noisy-label learning (Li et al., 2019; Wei et al., 2021). But we notice it has marginal gains in noisy-label FL. We think this is because the clients' local data are rare and the samples with wrong labels will have larger negative effects in the Mixup process. Intuitively, randomly mixing up the wrong-label samples with other samples will generate more noisy-label samples. Therefore, we propose a denoise Mixup loss to treat the noisy samples $\mathcal{N}_k$ and clean samples $\mathcal{C}_k$ differently. Specifically, for the detected noisy samples, we mix them with the samples from their corresponding class y_i^k (in Equation 10) to reduce the effects of wrong labels, while for the clean data, we adopt vanilla Mixup (i.e. randomly-mixed). Given a sample (x, y), we use the notation $(\tilde{x}, \tilde{y})$ to denote the corresponding-class-only Mixup strategy and the notation $(\hat{x}, \hat{y})$ to denote the randomly-mixed Mixup strategy. Thus, our denoise Mixup loss for client k is formulated as

$$\mathcal{L}_k^{mix}(\theta_k) = \frac{1}{|\mathcal{N}_k|} \sum_{(x_i^k, y_i^k) \in \mathcal{N}_k} \mathcal{L}_{mix}(\theta_k; (\tilde{x}_i^k, \tilde{y}_i^k)) + \frac{1}{|\mathcal{C}_k|} \sum_{(x_j^k, y_j^k) \in \mathcal{C}_k} \mathcal{L}_{mix}(\theta_k; (\hat{x}_j^k, \hat{y}_j^k)). \tag{11}$$

The denoise Mixup loss can improve the performance over vanilla Mixup and vanilla CE loss, and the result is in Table 6 of the appendix.

Prototypical Similarity Loss. Moreover, we consider the similarity of a noisy sample and its pseudo label's prototype is also a learning point. According to experimental evaluations (the left figure in Figure 9 of the appendix), we observe that the pseudo label precision is confident enough to reduce the gap between a noisy sample and its corresponding prototype. Thus, we devise a prototypical similarity loss for the noisy samples $\mathcal{N}_k$.

$$\mathcal{L}_k^{sim}(\theta_k) = \frac{1}{|\mathcal{N}_k|} \sum_{(x_i^k, y_i^k) \in \mathcal{N}_k} (1 - \cos(p_{k,y_i^k}, \mathcal{F}(\theta_k; x_i^k))). \tag{12}$$

Overall, the noise-resilience loss of client k is the sum of the two mentioned losses, formulated as:

$$\mathcal{L}_k^{sum}(\theta_k) = \mathcal{L}_k^{mix}(\theta_k) + \lambda_{sim}\mathcal{L}_k^{sim}(\theta_k), \tag{13}$$

where λ_{sim} is a hyper-parameter controlling the strength of $\mathcal{L}_k^{sim}$. Clients locally adopt SGD to minimize $\mathcal{L}_k^{sum}$ and update model parameters. The local training procedures are repeated for E epochs, and then the updated weights are sent to the server.

4.3 ROBUST GLOBAL AGGREGATOR

Learning with label noise has different training dynamics in the early and late. In the works of generalization and memorization (Shen & Sanghavi, 2019; Chatterjee, 2019; Chatterjee & Zielinski, 2020), it is found that in the early training period, generalization takes place that the neural networks learn the correct samples which have common patterns, while in the late training, the networks memorize the noisy data and fail in generalization.

In FL with noisy labels, we found the learning periods are also important in aggregation[1]. By a simple aggregation strategy switching, the overall performance can be further improved. We denote

[1]There is a previous work about critical learning periods in FL (Yan et al., 2021), but it is not about noisy-label learning.

T_{s_2} as the switching round for aggregation, and the learning periods can be divided into the early (before round T_{s_2}) and late (after round T_{s_2}). During the early period, where mostly the generalization takes place, we adopt the data-size-based aggregation as FEDAVG. In the late period, where bad memorization may occur, we cautiously aggregate clients' model parameters according to the noise levels. Hence, the robust global aggregator is:

$$\theta^{t+1} = \begin{cases} \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{N} \theta_k^t, & t < T_{s_2}, \\ \sum_{k=1}^{K} \frac{|\tilde{\mathcal{C}}_k|}{M} \theta_k^t, & t \geq T_{s_2}, \end{cases} \tag{14}$$

where $N = \sum_{k=1}^{K} n_k$ is the sum of local data sizes, and $M = \sum_{k=1}^{K} |\tilde{\mathcal{C}}_k|$ is the sum of local clean data sizes. The results in Figure 10 and Table 7 in the appendix validate that such a switching scheme can improve overall convergence.

Recall that in Section 4.1, we use switching round T_{s_1} to determine the time for applying pseudo labels to noisy samples. For simplicity, we set $T_s = T_{s_1} = T_{s_2}$. Generally, T_s is relatively small compared with the number of all rounds T (e.g. $T_s = 15,\ T = 100$).

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTING

Datasets and Models. Our experiments are conducted on three datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and Clothing1M (Xiao et al., 2015). We follow the existing works and apply Resnet-18 (He et al., 2016) for CIFAR-10, Resnet-34 for CIFAR-100, and Resnet-50 for Clothing1M.

Data Partition and Noise Distribution. (i) We adopt a general Non-IID data partition by Dirichlet distribution $q_j^k \sim Dirichlet(\alpha)$ to allocate a portion of q_j^k of the samples in class k to client j. We control the parameter α to adjust the degree of Non-IIDness. We consider α in {1, 0.7} for experiments. (ii) To generate real-world noisy labels in heterogeneous FL environments, we allocate a truncated Gaussian distribution $\chi_k \sim g(\chi_k; \mu, \sigma, a, b)$ to formulate noise level for each client. The limit is $a = 0$, and $b = 1$. We sample two groups of noise levels, a lower group is $\mu = 0.2, \sigma = 0.2$, and a higher noise level group is $\mu = 0.4, \sigma = 0.2$. We sample once and fix the two groups of noise ratios for experiments. We corrupt these datasets by two widely-used types of noisy labels: symmetric flipping (Yang et al., 2022; Ghosh et al., 2017) and pair flipping (Han et al., 2018). More details are in Appendix A.1.

Baselines. We compare FEDCNI with the following state-of-the-art methods in four groups: i) general federated learning optimization methods: FEDAVG (McMahan et al., 2017), FEDPROX (Li et al., 2020), FEDDYN (Acar et al., 2021) SCAFFOLD (Karimireddy et al., 2020); ii) a prototype based federated learning solution: FEDPROTO (Tan et al., 2022); iii) methods designed for label noise in centralized learning: CO-TEACHING (Han et al., 2018), DIVIDEMIX (Li et al., 2019), and we construct a distributed implementation and a combination with FEDAVG; iv) FL methods to tackle label noise without proxy datasets: FEDCORR (Xu et al., 2022), ROFL(Yang et al., 2022).

Implementation details. We use 20 clients fully participating in FL training in each round. We use the SGD optimizer with a learning rate of 0.01 and momentum of 0.5 in all experiments. The entire federated learning training process will last for 100 rounds to ensure convergence. We keep the number of local epochs $E = 5$ and the local batch size as 100 in all experiments. For the switching round, we set $T_s = 15$. The default confidence is set as $\tau = 0.5$, the hyper-parameter $\lambda_{sim} = 0.7$. We provide the hyper-parameters for baselines in Appendix A.1.

5.2 MAIN RESULTS

We compare FEDCNI with state-of-the-art methods in multiple noise types, noise ratios, and imbalance levels. The results of CIFAR-10/100 are shown in Table 1. Table 3 shows the results on the real-world dataset Clothing1M, which itself already contains real-world label noise. In summary, **FEDCNI achieves the best test accuracy in all noise settings tested on the datasets, especially at high noise levels**. For CIFAR-10, FEDCNI consistently outperforms all baselines except FEDPROTO by at least 5%. To compare noisy-label learning methods of centralized learning, we apply CO-TEACHING or DIVIDEMIX to each client.

Table 1: **Test accuracies (Top-1 %) on the CIFAR-10/100 dataset with symmetric and pair flipping noises.** Blue/**bold** fonts highlight the best baseline/our method.

Method	CIFAR-10								CIFAR-100							
	Symmetric				Pair				Symmetric				Pair			
	$\mu=0.2, \sigma=0.2$		$\mu=0.4, \sigma=0.2$		$\mu=0.2, \sigma=0.2$		$\mu=0.4\ , \sigma=0.2$		$\mu=0.2, \sigma=0.2$		$\mu=0.4, \sigma=0.2$		$\mu=0.2, \sigma=0.2$		$\mu=0.4\ , \sigma=0.2$	
	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7
FedAvg	79.94	77.95	64.89	63.28	80.58	81.10	67.98	60.42	48.61	47.62	36.35	37.63	52.54	53.18	38.43	37.62
FedProx	79.25	77.75	66.21	64.12	79.63	80.46	62.30	63.54	48.2	47.58	37.22	36.6	52.93	52.24	39.26	38.13
FedDyn	76.56	70.22	60.54	60.07	70.11	69.07	55.36	57.88	1.08	1.2	1.36	1.21	1.14	1.31	1.1	1.28
SCAFOLLD	79.15	77.87	63.79	66.29	80.29	79.26	61.12	61.84	49.45	47.73	36.79	36.46	54.01	51.16	38.92	39.01
FedCorr	79.02	78.18	64.02	61.16	76.24	78.47	55.72	63.07	37.74	39.02	37.09	38.53	52.77	50.07	40.21	39.91
RoFL	80.59	81.71	65.27	65.92	81.37	80.22	72.21	65.32	47.71	48.09	45.5	48.09	47.15	46.49	44.83	46.72
FedProto	82.11	80.87	72.16	71.93	80.09	82.36	61.67	69.74	55.83	55.2	45.84	44.78	59.78	58.11	45.29	44.28
Distributed Co-teaching	31.72	29.76	28.30	26.28	28.65	26.43	22.33	21.14	9.71	7.98	8.42	8.15	8.19	8.75	7.20	6.55
Distributed DivideMix	48.81	44.09	34.58	31.14	46.64	43.57	33.46	33.02	17.60	18.57	14.52	13.45	16.34	18.24	14.16	13.92
FedAVG+Co-teaching	45.95	29.78	39.85	32.63	42.93	35.73	30.43	22.14	16.09	14.37	9.47	8.48	15.73	13.7	9.73	8.52
FedAVG+DivideMix	80.23	79.68	65.17	60.72	74.15	76.13	54.31	54.19	39.28	37.18	30.06	28.69	45.25	44.37	35.12	27.05
Ours	**86.62**	**84.38**	**78.02**	**78.45**	**86.13**	**82.37**	**72.16**	**71.04**	**62.13**	**56.42**	**54.37**	**50.29**	**61.33**	**59.14**	**53.07**	**50.42**

In most cases, the method using DivideMix has higher accuracy and stability than Co-teaching. They all perform worse than federated learning baselines, which proves that a simple application of CL methods is not proper. Besides, FedAvg+Co-teaching has difficulty in converging when the number of communication rounds is limited. We also find that FedProto, which aggregates updates based on prototypes, also consistently surpasses the other baselines. *Prototype-based methods can thus be validated and can be more effective in noisy environments.* In CIFAR-100, Our FedCNI also shows effectiveness with 2% to 9% advantage. Our method shows robustness regardless of the noise generated by the symmetric-flip or pair-flip transformation matrices.

Table 2: Best test accuracy of different methods on CIFAR-10 with 4 groups different pair-flipping strategies.

Method\$(\mu, \sigma, \alpha_{Dir})$	$(0.2, 0.2, 1)$
FedAvg	84.40
FedProx	84.95
FedDyn	72.27
SCAFFOLD	84.87
FedCorr	77.16
RoFL	85.32
FedProto	86.45
FedVAG+DivideMix	83.63
Ours	**88.85**

We further investigate a condition that clients adopt different pair-flipping strategies in Table 2. Distinguished with the same mapping strategy for all clients, we divide clients into four groups randomly. In each group, clients apply the uniform pair-flipping strategy. The results show that the performances of all methods increase from 3% to 5%. Our method is still state-of-art with 2% to 10% advantages. We infer that the uniform pair-wise flip for all clients causes a global misunderstanding, while different strategies remix the labels randomly and enhance the model generalization.

Table 3: **Test accuracy on Clothing1M which naturally contains noise.**

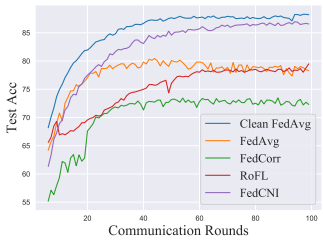
Dataset/ Method	FedAvg	FedProx	FedDyn	SCAFFOLD	FedCorr	RoFL	FedProto	Co-teaching	DivideMix	Ours
Clothing1M	68.34	69.85	70.55	69.36	72.4	73.31	70.52	69.83	70.1	74.26

5.3 Multi-dimension performances analysis

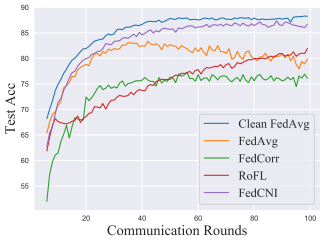
Convergence performance. As the results shown in Figure 2, we compare with FedCorr, RoFL, FedAvg, and FedAvg with clean data as an upper bound. Figure 2 (a, b) is conducted on the condition of $\mu = 0.2$, $\sigma = 0.2$, $\alpha = 1$ in two types of noise. We can observe that our FedCNI continues to rise rapidly in rounds about 20 to 40 where baselines have reached their convergence at lower accuracy. *We infer that the start timing of our noise-level re-weighting and pseudo labelling boosts the learning performance at these rounds.*

Noise Detection performance. We investigate the performance of noise detection. As shown in Figure 4 (a), we compare our method with FedCorr and RoFL, where FedCorr only detects for limited times instead of every epoch. The results show that **our method can outperform with 50% higher than RoFL, 70% higher than FedCorr.** Our noise detection also improves along with model learning. Besides, focusing precision is not comprehensive, because some clean samples may be misclassified into the noise set. Thus, Figure 4 (b) gives the corresponding results of recalls. Our method also shows 40% advantage over other baselines and also stabilizes with learning. We also present results among clients in Figure 4 (c, d) in FedCNI. Based on the difference between the average and the highest, we can find that most clients are above or around the average, and very few clients with extremely skewed data or high noise ratio are low.

Label correction performance. To verify the effectiveness of pseudo labelling, we further observe the average accuracy between the given pseudo labels and ground-truth labels across clients. The



(a) Symmetric-flipping



(b) Pair-flipping

Figure 2: **Convergence performance.** Experiments of representative methods on CIFAR-10, and upper bound FEDAVG with total clean data.

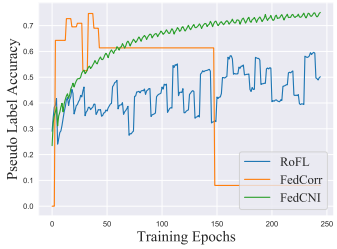
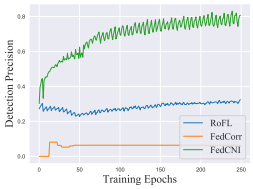
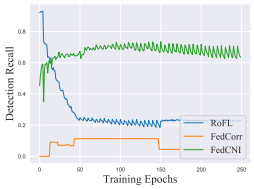


Figure 3: **Pseudo label accuracy performance.** Experiments to evaluate average pseudo label accuracy on CIFAR10



(a) Precision



(b) Recall

(c) Ours Precision

(d) Ours Recall

Figure 4: **Noise detection precision and recall.** Experiments to compare the ability of noise detection on federated learning with label noise baselines on CIFAR-10.

results in Figure 3 show that our prototypical local noise detection outperforms two federated noise learning baselines. **Our FEDCNI can stabilize at greater than 70% accuracy in label correction, which is 10% to 20% higher than baselines.** FEDCORR has high accuracy at the beginning, but only correct labels for few times instead of every epoch and round. Besides, due to the failure of small-loss method, two baselines may change the clean samples leading to unsatisfactory accuracy.

5.4 ABLATION STUDY

We conduct experiments to validate the effect of components in FEDCNI. We subtract each component to observe the accuracy changes. *All components help to improve accuracy where Mixup has a greater impact. The fusion of noisy samples and clean samples in the same class results in the bad memorization of the network.* If the pseudo label is incorrect, the diverge of fusion images can also be distinguished. Dynamic Confidence shows a smaller impact. We infer that it mainly helps to provide a threshold for pseudo-labelling to improve pseudo-label precision and converge speed.

Table 4: **Ablation study results**

Method	CIFAR-10 Test Accuracy(%)							
	Symmetric				Pair			
	μ=0.2, σ=0.2		μ=0.4, σ=0.2		μ=0.2, σ=0.2		μ=0.4 , σ=0.2	
	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7	α=1	α=0.7
FEDAVG	78.69	76.32	65.33	64.15	81.56	81.66	68.4	60.11
Ours w/o Dynamic Confidence	85.02	84.09	76.94	77.92	85.94	81.59	71.97	70.61
Ours w/o Mixup	82.46	81.57	73.84	74.52	82.17	78.54	69.39	67.52
Ours w/o Distance Loss	86.5	84.96	78.2	76.28	85.6	82.37	70.86	71.36
Ours w/o Re-weight Aggregation	84.93	82.13	75.69	75.06	84.03	79.68	68.81	68.49
Ours	**86.62**	**85.76**	**78.37**	**78.1**	**86.52**	**82.65**	**72.04**	**71.54**

6 CONCLUSION

In this paper, we propose the FEDCNI, a novel federated learning method to tackle the label noise in class-imbalanced data. To deal with the challenges, we present a bi-level solution that the clients cautiously learn from noisy data, and the server critically aggregates from noisy clients. In FEDCNI, the noise-resilience local solver use a prototypical method to detect and then correct local imperfect annotations with dynamic confidence. While the robust global aggregator realizes the switching between the data-size weighted aggregation and the noise-level re-weighted aggregation in different learning periods. Extensive experiments demonstrate the effectiveness of the proposed mechanism.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *Proceedings of International Conference on Learning Representations*, 2021.

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 *Proceedings of International conference on machine learning*, pp. 233–242. PMLR, 2017.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6912–6920, 2021.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *Proceedings of International Conference on Learning Representations*, 2019.

Satrajit Chatterjee and Piotr Zielinski. Making coherence out of nothing at all: measuring the evolution of gradient alignment. *arXiv preprint arXiv:2008.01217*, 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.

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 (CVPR)*, pp. 10072–10081, June 2022.

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.

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. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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 *Proceedings of International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5132–5143, 13–18 Jul 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

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

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *Proceedings of IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978, 2022.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 429–450, 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Shashi Raj Pandey, Nguyen H. Tran, Mehdi Bennis, Yan Kyaw Tun, Aunas Manzoor, and Choong Seon Hong. A crowdsourcing framework for on-device federated learning. *IEEE Transactions on Wireless Communications*, 19(5):3241–3256, 2020.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5739–5748. PMLR, 09–15 Jun 2019.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 1, pp. 3, 2022.

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

Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*, 2021.

Tong Wei, Jiang-Xin Shi, Yu-Feng Li, and Min-Ling Zhang. Prototypical classifier for robust class-imbalanced learning. In *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 44–57. Springer, 2022.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Jingyi Xu, Zihan Chen, Tony Q.S. Quek, and Kai Fong Ernest Chong. Fedcorr: Multi-stage federated learning for label noise correction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Gang Yan, Hao Wang, and Jian Li. Critical learning periods in federated learning. *arXiv preprint arXiv:2109.05613*, 2021.

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.

Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. *IEEE Intelligent Systems*, 37(2):35–43, 2022.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021a.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021b.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of International Conference on Learning Representations*, 2018.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Datasets and Models. Our experiments are conducted on three datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and Clothing1M (Xiao et al., 2015). Clothing1M (Xiao et al., 2015) is a real-world dataset that comprises 1 million training apparel images taken from 14 categories of online shopping websites. It has real-world noisy labels.

Table 5: Summary of data sets used in the experiments.

Dataset	CIFAR-10	CIFAR-100	Clothing1M
# of Training $\mathcal{D}_{train}$	50,000	50,000	1,000,000
# of Testing $\mathcal{D}_{test}$	10,000	10,000	10,000
# of classes	10	100	14
image size	32 ×32	32 ×32	256 ×256

We follow the existing works and apply Resnet-18 for CIFAR-10, Resnet-34 for CIFAR-100, and Resnet-50 for Clothing1M (He et al., 2016).

Noise Distribution. We consider the label noise in real-world is heterogeneous across clients. We have χ_k as the noise level of a client which equals $\frac{|\mathcal{N}_k|}{|\mathcal{D}_k|}$. We assume the client's noise level χ_k initially has a Gaussian distribution $\chi_k \sim \phi(\chi_k) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(\chi_k-\mu)^2}{2\sigma^2})$, where μ is the mean and σ^2 is the variance. Then we add a limit (a, b) to form a truncated Gaussian distribution as:

$$\chi_k \sim g(\chi_k; \mu, \sigma, a, b) = \frac{\frac{1}{\sigma}\phi(\frac{\chi_k-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma}) - \Phi(\frac{a-\mu}{\sigma})}, \tag{15}$$

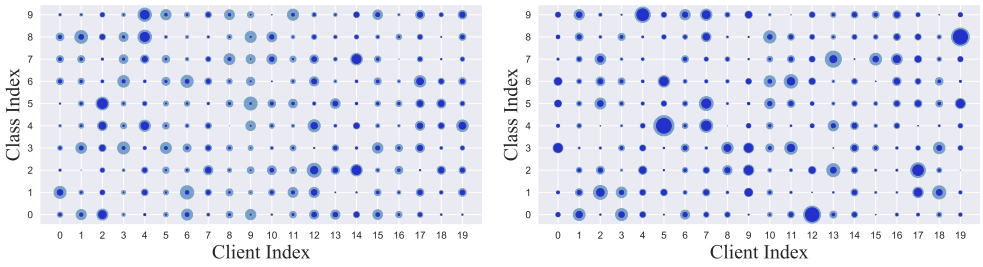
where $\Phi(\cdot)$is the cumulative distribution function. We sample two groups of noise level based on truncated Gaussian distribution where we set $a = 0$, $b = 1$. We set a lower group as $\mu = 0.2, \sigma = 0.2$, and a higher noise level group as $\mu = 0.4, \sigma = 0.2$. We sample once and fix the two groups of noise ratios for experimenting. We consider the two types of noisy labels:

- **Symmetric flipping:** All samples will be mislabeled as other labels with the same probability (Yang et al., 2022; Ghosh et al., 2017). For each client, We sample a proportion of data, and randomly replace their labels of a certain percentage of the training data with all possible labels.
- **Pair flipping:** A kind of asymmetric noise where mislabeled samples are annotated within very similar classes (Han et al., 2018). (e.g. deer→horse, dog⟷ cat). Our most experiments are conducted on the same mapping strategy for all clients. We also investigate the condition of different pair flipping strategies between clients.

Data Partition. In a realistic FL scenario, local datasets are usually Non-IID and frequently imbalanced. We adopt a heterogeneous data partition by Dirichlet distribution $q_j^k \sim Dirichlet(\alpha)$ to allocate a portion of q_j^k of the samples in class k to client j. We control the parameter α to adjust the degree of Non-IIDness. To generate the Non-IID data distributions on CIFAR10/100, we consider Dirichlet parameter α in {1,0.7} in experiments.

Baselines. We compare FEDCNI with the following state-of-the-art methods in four groups: i) general federated learning optimization methods: FEDAVG (McMahan et al., 2017), FEDPROX (Li et al., 2020), FEDDYN (Acar et al., 2021) SCAFFOLD (Karimireddy et al., 2020); ii) a prototype based federated learning solution: FEDPROTO (Tan et al., 2022); iii) methods designed for label noise in centralized learning: CO-TEACHING (Han et al., 2018), DIVIDEMIX (Li et al., 2019); iv) FL methods to tackle label noise without proxy datasets: FEDCORR (Xu et al., 2022), ROFL(Yang et al., 2022).

Implementation details. We use 20 clients fully participating in FL training in each round. We set SGD optimizer with learning rate of 0.01, and momentum of 0.5 in all experiments. The entire federated learning training process will last for 100 rounds to ensure convergence. We always apply



(a) $\alpha = 1, \mu = 0.2, \sigma = 0.2$, Symmetric-flip (b) $\alpha = 0.7, \mu = 0.4, \sigma = 0.2$, Pair-flip

Figure 5: **The example of the sizes and noises of samples per class allocated to each client.** The size of the circle represents the number of samples. The lighter color shows the size of clean samples, and the darker color denotes the noisy sample quantity.

5 local epochs and 100 local batch sizes. For the end of the early phase, we use $T_s = 15$. The default confidence is set as $\tau = 0.5$, the loss parameter $\lambda_{sim} = 0.7$.

Hyper-parameters. For FEDPROX (Li et al., 2020), we set the proximal term μ as 0.01. For FEDDYN (Acar et al., 2021), we tune the penalty coefficient α over {0.1,0.01,0.001},and display the results of 0.1. DIVIDEMIX (Li et al., 2019) has some epochs for warm-up, we let it be 5 here. We follow the default $p_{threshold} = 0.5$ to detect noisy samples as their original code. For fairness, in ROFL(Yang et al., 2022), we set the parameter T_{pl} of the round to start using global guided pseudo labelling as 15, which equals our T_s to adjust aggregation and start pseudo labelling. FEDCORR (Xu et al., 2022) has three phases of learning, we form them to fulfil the uniform 100 global rounds as iteration1 = 10, rounds1=20, and rounds2=70. There are some hyper-parameters not mentioned, which are not needed to be adjusted. We all follow the original settings.

A.2 ADDITIONAL DETAILED EXPERIMENTS

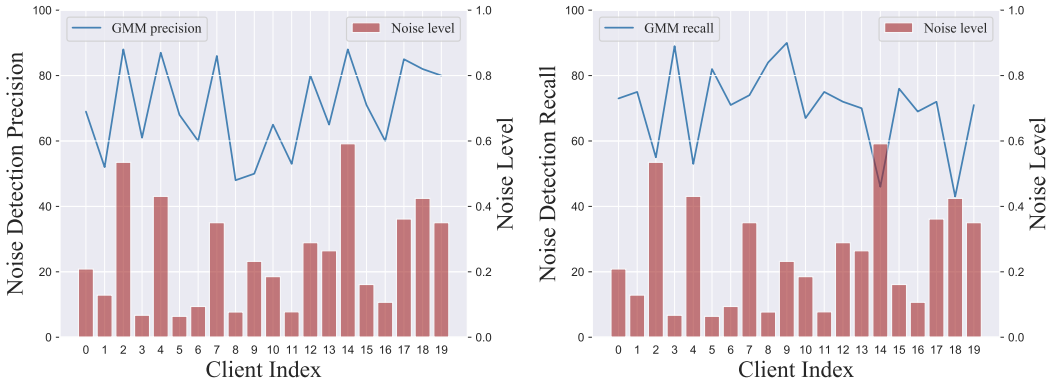
Effects of data conditions on noise detection. We evaluate our method in three more specific dimensions: noise level, data size and class imbalance degree. The results shown in Figure 6, 7, 8 provide the relation between noise detection precision/recall and one of dataset's attributes. Firstly, our FEDCNI achieves a confident noise detection accuracy in the view of each client in Figure 6 (a). We have shown another statistical view in Figure 4. Based on the average accuracy that is above 70%, and the tolerable worst performance, we can conclude that our prototypical noise detection is robust in a noisy and skewed data condition. The performance of recall in Figure 6 (b) also shows that not many clean samples are regarded as noisy samples. By observing Figure 6, 7, 8, we infer that the trend between noise level and noise detection accuracy is equally significant with the trend between imbalance degree and noise detection accuracy. And the data size has a minor impact on detection accuracy, only if an extreme small quantity.

Effects of Mixup strategies. In the Section 4.2, we propose a denoise Mixup. There are two significant baselines: vanilla Mixup and classic cross-entropy loss. We validate our proposed denoise Mixup loss in Table 6. The mixture of noisy samples and clean samples with the same class enhances our methods with 1.4% and 4% advantages over baselines.

Table 6: **Test accuracies on different Mixup strategies.**

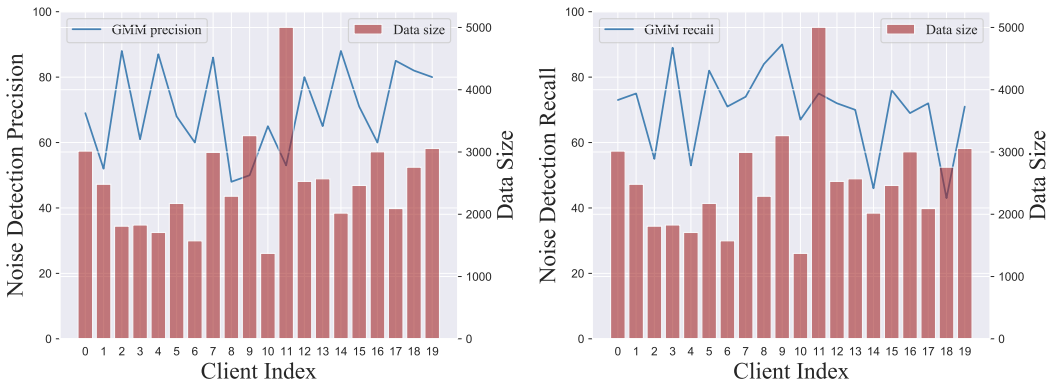
Dataset/ Method	FEDCNI	FEDCNI with vanilla Mixup	FEDCNI with CE loss
CIFAR10	86.62	85.23	82.46

Effects of the switching round T_{s1}. As motioned in Equation 10, we evaluate the effect of hyper-parameter T_{s1} in Figure 9. The switching round T_{s1} is used to control the starting round of pseudo-labelling. We record the accuracy of pseudo-labelling but do not apply new labels to noisy samples before T_{s1} in Left Figure 9. We can observe that the accuracy of pseudo labelling is relatively lower than 65% before the switching round $T_{s1} = 15$. It gradually stabilizes to be above 70% after the utilization of pseudo labels. We suggest that in the early phase, the accuracy of pseudo labeling is not



(a) Precision (b) Recall

Figure 6: **Noise detection precision and recall on clients' noise levels.**



(a) Precision (b) Recall

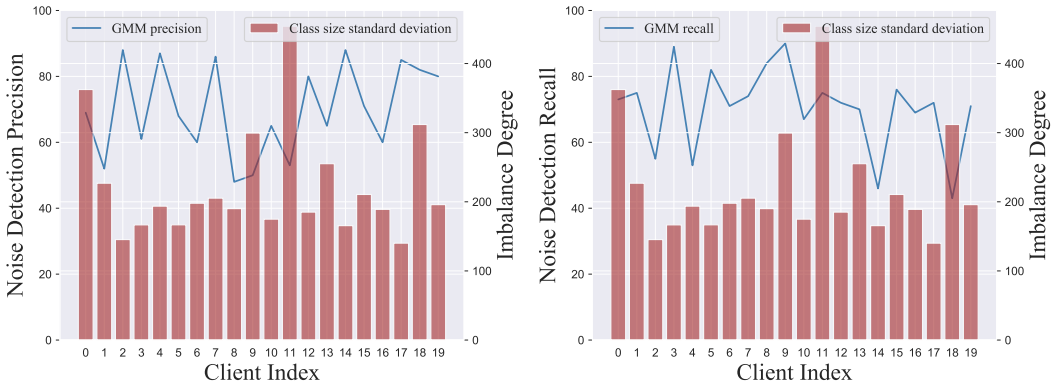
Figure 7: **Noise detection precision and recall on clients' data sizes.**

confident enough, so we give pseudo labels after a better performance to enhance the generalization of the model. We also compare $T_{s1} = 15$ with $T_{s1} = 0$ in Right Figure 9, where the pseudo labeling begins since the training starts. We can find the converge speed and test accuracy of $T_{s1} = 0$ are both worse than a confident switching round.

Table 7: **Test accuracies in terms of aggregation strategy in different learning periods.**

Dataset/ Method	FEDCNI	FEDCNI-data-size	FEDCNI-noise-level
CIFAR10	86.62	84.92	85.04

Effects of the aggregation strategy. We present a switching strategy for federated aggregation in Section 4.3. To verify the effectiveness of our aggregator, we compare with (i) based on the sum of local data sizes in the whole process;(ii) based on the sum of local clean data sizes in the whole process. The results in Table 7 show that our aggregation strategy outperforms the two baselines with 1% to 2% advantages. Such a switching strategy is indeed effective. We can also observe the learning curve after the switching is always higher than the stable aggregation strategies in Figure 10.



(a) Precision (b) Recall

Figure 8: **Noise detection precision and recall on clients' imbalance degrees.**

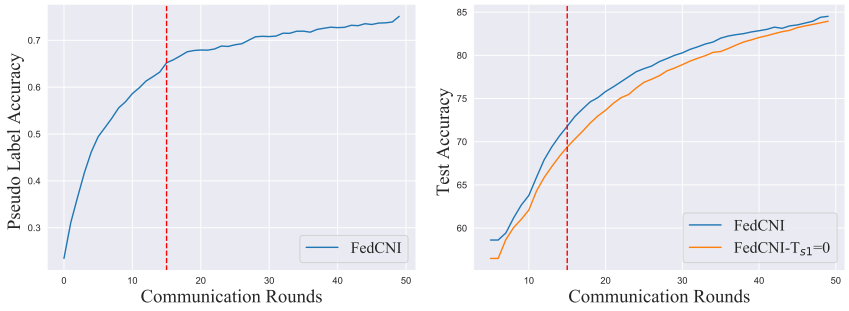


Figure 9: **Left:** the accuracy of pseudo labelling. **Right:** the training performances of whether utilizing pseudo label since start. The red dotted vertical line refers to the switching round T_{s_1} in FEDCNI.

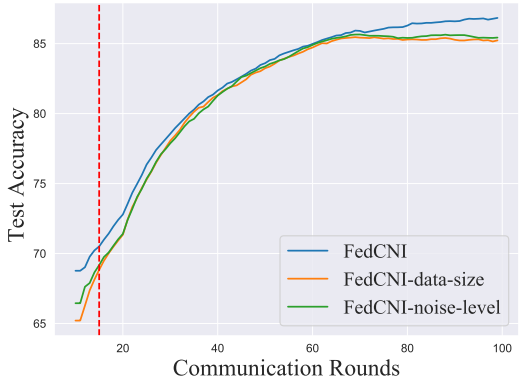


Figure 10: **Training performances in terms of aggregation strategy in different learning periods.** The red dotted vertical line refers to the switching round T_{s_2} in FEDCNI. FEDCNI-data-size/FEDCNI-noise-level refers to using data-size-based/noise-level-based aggregation throughout training in FEDCNI.