Federated Noisy Client Learning

Kahou Tam®, Li Li®, Bo Han®, Senior Member, IEEE, Chengzhong Xu®, Fellow, IEEE, and Huazhu Fu®, Senior Member, IEEE

Abstract—Federated learning (FL) collaboratively trains a shared global model depending on multiple local clients, while keeping the training data decentralized to preserve data privacy. However, standard FL methods ignore the noisy client issue, which may harm the overall performance of the shared model. We first investigate the critical issue caused by noisy clients in FL and quantify the negative impact of the noisy clients in terms of the representations learned by different layers. We have the following two key observations: 1) the noisy clients can severely impact the convergence and performance of the global model in FL and 2) the noisy clients can induce greater bias in the deeper layers than the former layers of the global model. Based on the above observations, we propose federated noisy client learning (Fed-NCL), a framework that conducts robust FL with noisy clients. Specifically, Fed-NCL first identifies the noisy clients through well estimating the data quality and model divergence. Then robust layerwise aggregation is proposed to adaptively aggregate the local models of each client to deal with the data heterogeneity caused by the noisy clients. We further perform label correction on the noisy clients to improve the generalization of the global model. Experimental results on various datasets demonstrate that our algorithm boosts the performances of different state-of-the-art systems with noisy clients. Our code is available at https://github.com/TKH666/Fed-NCL.

Index Terms—Federated learning (FL), label noise, noisy client, noisy learning.

I. INTRODUCTION

LOCAL clients, such as personal devices, financial institutions, or hospitals, have access to a wealth of private data. However, due to data privacy concerns, security limitations, and device availability, it is impractical to collect and store all the data from the local clients at the server center and conduct centralized training. To ensure data privacy in the learning procedure, federated learning (FL) is recently proposed as a workflow that enables local clients to collaboratively train a shared global model without sharing local private data [1], [2]. The workflow of a typical FL framework can be mainly divided into the following two stages: 1) the client performs local training independently based on its local data, and the trained local model is sent back to a central server and 2) the central server then aggregates the local gradient updates and generates the global model. This process iterates until a certain accuracy level of the learning model is reached. In this way, an accurate predictive model can be obtained while the user data privacy is effectively protected, as the local training data are not accessed directly. Thus, FL has been widely adopted to support different application scenarios with high requirements of data security and privacy, such as smart retailers collecting user purchasing records [3], hospitals containing a multitude of patient data for predictive health care [4], and financial institutions storing highly personal data for risk prediction [5].

Despite all its promise, one critical obstacle exists for FL to be viable in real-world scenarios. In general, the performance of machine learning (ML) highly depends on the data quality of the participating clients [6]. However, collecting the high accurate labeled data can be both time-consuming and expensive. Even experienced individuals find the data annotation process extremely complex. In real-world human-annotated datasets, noisy labels can account for anywhere between 8.0% and 40.2% [7], [8]. It is challenging to ensure that training data labels on each client are clean, as it is impossible to guarantee perfect annotation by all the clients. Training deep learning models with noisy data (e.g., feature and label noise) can lead to overfitting on corrupted data, thereby reducing the model’s generalization performance [9]. In the context of FL, where the global model depends on a large number of clients, the noisy data issue becomes more pronounced. Clients may have entirely different standards for collecting their local data, and some may be nonexpert or uncertain sources, leading to unreliable labels that can harm the global aggregated model. Unfortunately, standard FL aggregation methods such as FedAvg [10] and FedProx [11] overlook the noisy client issue and develop the global model directly based on the trained model from each participating client. As a result, the overall performance of FL can be negatively impacted by noisy clients.

Recently, many ML techniques have been developed to handle noisy data [12], [13] in the centralized setting. The state-of-the-art noisy learning methods have shown significant improvement by reweighting training samples [14] or applying...
a disagreement constraint between different networks/training strategies [15], [16], [17]. However, these noisy learning methods cannot be applied in FL workflow directly. First, the local training data are inaccessible to the central server. Thus, it cannot remove/reweight the noisy data during the global model aggregation. Second, different from the single dataset in standard ML, FL has several independent local clients, which have different noisy data distributions. This makes it difficult to estimate the unified noise transition matrix used in noisy learning methods [14]. Third, some clients (e.g., battery-powered mobile devices) in FL may have limited computing resources, so that the specific noise adaptation layer [18] or additional collaborative network [17] cannot be satisfied.

Training the collaborative model with noisy label data in FL has recently gained more attention. Some methods [19], [20] use the extra clean data to identify the clients with noisy labels and mitigate their negative impact by reweighting during the global model aggregation. Yang et al. [21] first select the confident data to train the local model and perform label correction on noisy label data. However, these existing methods are still not effective and stable in the real-world setting [22] for the following reasons. First, from the perspective of noisy clients, they assume all the clients have the same noise degree instead of heterogeneous noise distribution, which is not practical in the real-world scenario. Second, they rely on the benchmark data to detect the clients with noisy label data and select the confident training data of noisy clients, which is hard to satisfy in the real-world scenario since it restricts the task generalization of FL. Third, most methods only consider the effectiveness without understanding how the noisy clients' local model affects the global model in FL. Thus, there is an urgent need for a framework that can effectively identify noisy clients and conduct robust learning without relying on benchmark data in FL.

In this article, we first conduct an analysis of the issue of noisy clients in FL, and model the noise levels among clients with two distributions, namely, the Bernoulli and truncated Gaussian distributions, to better emulate the heterogeneous noisy scenarios that are present in real-world federated systems. Next, we aim to identify the ways in which the presence of noisy clients affects the global model in FL and conduct an experimental study on the performance of the global model and convergence rate in the presence of noisy clients. Our findings indicate that the inclusion of noisy data from participating clients can have a negative impact on both the convergence and performance of the aggregated global model in FL. Furthermore, we delve deeper to investigate the effects of noisy clients on each layer of the global model in FL. To do so, we use the centered Kernel alignment (CKA) [23] to measure the output feature similarity of each layer between the noisy clients and global model. We observe that noisy clients can induce greater bias in the deeper layers of the global model than in the former layers when using FedAvg [10] to directly aggregate all the clients’ models.

Based on the above observation, we propose Fed-NCL, a federated noisy client learning framework, which effectively tackles the noisy clients in FL without relying on the benchmark data. In Fed-NCL, we detect noisy clients through modeling the reliability score distribution among the participating clients. The reliability score jointly considers each client’s local model divergence with global model and its data quality. Based on the observation that the deeper layer’s feature extraction ability in the noisy clients’ model heavily declined, a noise-robust layerwise aggregation is designed to aggregate the feedback model from each client by considering each layer’s divergence and guiding the global model update’s direction like clean clients. To fully use the local data from noisy clients to improve the generalization of the global model, we further perform the label correction on the noisy clients. In summary, the main contributions of this article are as follows.

1) We present the first systematic and empirical study on FL with noisy label data and analyze the impact of the hidden representations of different layers of deep neural networks (DNNs) trained with FedAvg on noisy label data.
2) We propose two types of noisy label scenarios in the real-world federated system by modeling the noise level among the clients with two distributions (e.g., Bernoulli and truncated Gaussian distributions).
3) Our systematic study on FL with noisy clients reveals that (1) the noisy clients could negatively impact the convergence and performance of the global model in FL, and (2) the noisy clients can induce greater bias in the deeper layers than the former layers of the global model.
4) We propose Fed-NCL, which dynamically identifies the noisy clients by reliability score without making any assumptions on the noisy clients. The reliability score measures each client’s data quality and model divergence with the global model. Then we develop a robust layerwise aggregation methodology, which mitigates the negative impact from the noisy clients in terms of layers. The label correction is dynamically performed in the noisy clients to improve the generalization capabilities of the global model.
5) Experimental results on various datasets demonstrate that our algorithm can boost the robustness and performances of different deep learning backbones in FL systems against noisy clients.

II. RELATED WORKS
A. Deep Learning With Noisy Label
Several methods have been designed to mitigate the impact of noisy data in deep learning [13], [14], [15], [16], [17], [18]. Goldberger and Ben-Reuven [18] proposed to model the data noise with an additional softmax layer during the training process. Menon et al. [14] reweighted the importance of the training samples to reduce the impact of noisy labels. Jiang et al. [16] designed a collaborative learning framework (MentorNet) to introduce a pretrained mentor network that directs the training process. In addition, Han et al. [17] also
proposed to maintain two networks, where each network selects the training samples that have a small loss and passes them as input to its peer network to continue the training process. However, these existing approaches cannot be directly adopted in FL scenarios for the following two reasons. First, to preserve data privacy, the central server cannot access the raw training data located on each client. Second, the computational overhead caused by introducing an extra network in the training process can highly impact the energy efficiency and user experience for battery-powered clients in FL (e.g., smartphones and wearable devices).

B. FL With Noisy Label

The existing works about FL with noisy labels can be summarized into the following three main categories based on the cleansing granularity: 1) client-level methods; 2) data-level methods; and 3) hybrid methods. The client-level cleansing methods usually detect noisy clients using extra benchmark data and reduce the negative impact of clients with noisy labels. FOCUS [19] exploits the benchmark data stored in the server to evaluate all the participant clients’ models to adjust their weight of global model aggregation. It allows for adjustments to be made to the weight of global model aggregation based on the quality of the client’s model. The data-level cleansing methods have developed following the improvement of noisy labels’ data detectors in centralized learning. An intriguing observation regarding deep models is that they possess the ability to memorize simpler instances initially, and subsequently adjust to more complex instances as the training epochs progress [9]. Based on this, RoFL [21] selects the sample with small loss as confident data to train and create local centroids and exchange them between clients and servers. DS [20] selects the confident samples by estimating the sample losses’ similarity between the client’s training data and benchmark data from the server. However, these works have limitations in terms of task generalization and privacy since the centralized label noise detector methods cannot be directly applied in FL. The hybrid cleansing methods integrate client and client-side data cleansing to address noisy clients in FL. For instance, FedCorr [24] identifies noisy clients and labels using model prediction subspaces and per-sample losses. It then fine-tunes the global model on clean clients and corrects noisy labels on noisy clients. It has a limitation in that it assumes the participating clients have enough clients with clean labels to first train the global model, which is hard to be satisfied in real-world applications. According to the above definition, Fed-NCL belongs to the client-level methods’ category, as it focuses on addressing noisy clients through client-level mechanisms.

III. PROBLEM STATEMENT

Several label noise distributions have been effectively modeled for standard ML [12], [25], [26], e.g., uniform or nonuniform random label noise. However, the noise scenario in FL is different from the label noise in standard ML. The noise modeling of standard ML only focuses on a single training dataset, while FL has multiple clients with their own private datasets. Thus, we focus on the noise distribution among all the participating clients. In this section, we present the problem statement of FL with noisy clients and model the noisy clients with different distributions.

A. FL With Noisy Client

In this article, we consider the $K$-class image classification, which is a representative supervised learning task and can be formulated to learn the mapping function $f(x; \Theta)$ from a set of training examples $D = \{(x_i, y_i)\}_{i=1}^N$ with $y_i \in \{0, 1\}^K$ being the ground-truth label in a one-shot manner corresponding to $x_i$. In deep learning, $f(x; \Theta)$ is a network and $\Theta$ represents the model parameters. The parameter $\Theta$ minimizes the empirical risk $\mathcal{R}(f)$, as

$$\mathcal{R}(f) = \sum_{(x_i, y_i) \in D} l(f(x_i; \Theta), y_i)$$

where $l(\cdot)$ is a loss function (e.g., cross-entropy (CE) loss or mean squared error).

In the FL workflow, as data labels are corrupted in the training data of the participating clients, we aim to train the model from noisy clients. Specifically, a certain client $c$ is provided with a noisy training dataset $D^c = \{(x_i^c, \tilde{y}_i^c)\}_{i=1}^N$ where $\tilde{y}_i^c$ is a noisy label. Hence, following the standard training procedure, a mini-batch $B^c_t = \{(x_i^c, \tilde{y}_i^c)\}_{i=1}^b$ comprising $b$ samples is obtained randomly from the noisy training dataset $D^c$ of client $c$ at time $t$. Subsequently, the local model parameter $\Theta^c_t$ of client $c$ at time $t$ is updated along the descent direction of the empirical risk on mini-batch $B^c_t$

$$\Theta^c_{t+1} = \Theta^c_t - \eta^c \nabla \sum_{(x_i^c, \tilde{y}_i^c) \in B^c_t} l(f^c(x_i^c; \Theta^c_t), \tilde{y}_i^c)$$

where $\eta^c$ is a specified learning rate in client $c$. Thus, the risk minimization process is no longer noise-tolerant because of the loss computed by the noisy labels in the local training process for the noisy clients.

In a training round, participating clients send their updated model parameters to the central server after completing the local training. The central server then aggregates the global model $\Theta^G$ by federated averaging (FedAvg) [10], as

$$\Theta^G = \frac{1}{C} \sum_{c=1}^C \Theta^c$$

where $C$ represents the total amount of local clients in the current training round, and $\Theta^c$ represents the model parameter of client $c$ in the current training round. However, some of the participating clients are noisy clients, which means that some of the local models have memorized corrupted labels. Thus, directly aggregating these models using (3) leads the update of the global model to a divergent direction, and the generalization of the global model is severely impacted correspondingly. Hence, mitigating the negative effects of noisy clients is critical to make FL viable in practice.
B. Modeling Noisy Client

In this section, we discuss how to model the scenario of a noisy client with two types of distributions, i.e., Bernoulli and truncated Gaussian distributions. Then we also investigate how these two noise distributions impact the global model in FL.

1) Case 1: Bernoulli Distribution: In this case, we use Bernoulli distribution to model the noisy clients across the participants in a federation. The Bernoulli distribution is a discrete distribution having two possible outcomes labeled by \( n = 0 \) and \( n = 1 \), in which \( n = 1 \) (the training data of a specific client are clean) occurs with probability \( p \) and \( n = 0 \) (the training data of a specific client are noisy) occurs with probability \( q = 1 - p \), where \( 0 < p < 1 \). The probability density function \( P_B \) is represented as

\[
P_B(n) = p^n(1 - p)^{1 - n}.
\]

It is important to note that in this case, we assume that if a client is a noisy client \( (n = 0) \), the labels of all the training data within it are corrupted (e.g., some sensors on certain mobile clients malfunction and generate corrupted training data). Otherwise, if a client is a clean client \( (n = 1) \), the labels of all the training data have true labels. For the reason that whether the data collection component in client (e.g., sensors on mobile devices) malfunctions or not is entirely independent of each other and follows the Bernoulli process, and thus, Bernoulli distribution [27] is adopted in this case.

2) Case 2: Truncated Gaussian Distribution: In this case, the local training data of all the clients in FL may have labels corrupted to different degrees (e.g., different hospitals may collect data with various qualities). We use a truncated Gaussian distribution [28] to model the percentage of corrupted labels within the training data across different clients. Specifically, the truncated normal distribution is derived from a normally distributed random variable by giving the random variable the lower and upper bounds. Specifically, suppose \( X \) has a normal distribution with mean \( \mu \) and variance \( \sigma^2 \) and lies between \( (a, b) \). Then \( X \) follows a truncated normal distribution conditioned on \( a < X < b \). The probability density function \( P_G \), for \( a \leq x \leq b \) can be represented as follows:

\[
P_G(x; \mu, \sigma, a, b) = \frac{1}{\sigma} \frac{\phi\left(\frac{x - \mu}{\sigma}\right) - \Phi\left(\frac{b - \mu}{\sigma}\right)}{\Phi\left(\frac{b - \mu}{\sigma}\right) - \Phi\left(\frac{a - \mu}{\sigma}\right)},
\]

where \( \phi(\cdot) \) is the probability density function of the standard normal distribution, and \( \Phi(\cdot) \) is its cumulative distribution function. \( x \) is defined as the noise degree for a certain client, which can be represented as \( x = (n_{\text{noisy}}/n_{\text{clean}} + n_{\text{noise}}) \). \( n_{\text{noise}} \) is the amount of noisy data, while \( n_{\text{clean}} \) represents the amount of clean data within a certain client. Moreover, \( x \) is bounded between 0 and 1.

3) Impact of Two Noise Distributions: To investigate how these two noise distributions impact the performance of global model in FL, we perform experiments on the CIFAR10 [29] dataset using VGG16 [30]. Since the original dataset is clean, we manually inject noise using the symmetric noise approach [13]. This involves corrupting the true label using a label transition matrix \( T \). The definition of transition matrix \( T \) is as follows:

\[
T = \begin{bmatrix}
1 - \epsilon & \frac{\epsilon}{n-1} & \cdots & \frac{\epsilon}{n-1} \\
\frac{\epsilon}{n-1} & 1 - \epsilon & \cdots & \frac{\epsilon}{n-1} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\epsilon}{n-1} & \frac{\epsilon}{n-1} & \cdots & 1 - \epsilon
\end{bmatrix}
\]

where \( n \) is the number of the class, and \( \epsilon \) is the noise level. \( T_{ij} \) denotes the probability of the true label \( i \) being flipped into a corrupted label \( j \). The true labels are flipped into other labels with equal probability. Fig. 1 shows the test accuracy versus the communication rounds for VGG16 on CIFAR10 in the above two cases.

4) Bernoulli Distribution: For noisy clients following the Bernoulli distribution, we test two different scenarios: \( p = 0.7 \) and \( p = 0.8 \). Fig. 1(a) shows the test accuracy with different communication rounds. As can be seen, the training process can smoothly converge in a few communication rounds (e.g., before round 150 in this case). However, the convergence process slows down prominently when \( p = 0.8 \) (20% of the clients are noisy clients). In addition, the situation gets even worse when the percentage of noisy clients increases (e.g., \( p = 0.7 \)). This is because the local models located at the noisy client try to memorize the corrupted data in the local training process. In addition, during the model aggregation process, these local models will guide the update of the collaborative model in a divergent direction.

5) Truncated Gaussian Distribution: For noisy clients following the truncated Gaussian distribution, we investigate the influence of noisy clients whose noise follows the truncated Gaussian distribution on the FL process. We focus on two specific scenarios: \((\mu = 0.3, \sigma = 0.45)\) and \((\mu = 0.3, \sigma = 0.4)\) and evaluate the test accuracy throughout the entire training process. The obtained results are illustrated in Fig. 1(b). Our findings indicate that the inclusion of clients with truncated Gaussian distributed noise has a detrimental effect on the FL process. Moreover, we observe that the impact on the global model increases as the standard deviation of the distribution increases. It is noteworthy that in the presence of clients with truncated Gaussian distributed noise, the performance of the global model is unstable. This implies that convergence of the global model is hindered more in the presence of noisy clients following the truncated Gaussian distribution, compared with the case where clients’ noise follows the Bernoulli distribution.
IV. CLOSE LOOK AT NOISY CLIENT

In this section, we investigate the impact of noisy clients on different layers of the global model in the standard FL procedure [10]. First, we quantify the negative impact of noisy clients in terms of the representations learned by the model. Next, we compare the feature similarity between the global model and clean/noisy clients to demonstrate the feature variance caused by noisy clients. Furthermore, we present the results of our analysis based on a quantitative study with noisy clients in the FL setting.

To explore the impact of noisy clients, we conduct a quantitative analysis to examine how noisy labels affect the representations of different layers for both clean and noisy clients. We train the global model on ten clients, with data partitioned according to the i.i.d setting. To observe the effects of noisy clients, we design five clients as noisy clients and manually injected 50% symmetric label noise into their training data. The remaining clients have clean labels. To quantitatively demonstrate the effect of noisy clients on each layer, we measured the extraction ability of each layer’s representations by computing the similarity between the representations from the same layer of different clients’ local models, using the CKA [23]. The CKA is a similarity metric used to compare the representations of two neural networks. It measures the similarity between the representations of two neural networks by computing the inner product between their centered kernel matrices. CKA has been shown to be effective in comparing the representations of neural networks trained on different datasets or with different architectures [31], [32]. It can determine how much information is preserved between the representations, enabling insights into how much each network has learned in common. In our experiments, we use linear CKA to calculate the similarity of the output features between the two models. Given the same dataset \( D_{clean} \), feature matrix \( Z_1 \), \( Z_2 \) can be extracted by two models, respectively.

Then we can calculate the linear CKA similarity between two representations \( X \) and \( Y \) as

\[
\text{CKA}(X, Y) = \frac{||X^T Y||_F^2}{||X^T X||_F^2 ||Y^T Y||_F^2}.
\]

(7)

The CKA similarity score is between 0 (not similar) and 1 (identical) to show the output feature similarity of the same layer across two local models. We train the global model using the FedAvg [10] for 100 communication rounds, and the clients locally train for ten epochs at each round. Fig. 2 shows the pairwise CKA features’ similarity score of the four different layers among the clients’ local models and the global model. The selected layers are the model’s first, fourth, seventh, and last layers (classifier layer). Interestingly, we find that the deeper layers in the model have lower CKA feature similarity scores between noisy and clean clients. This observation indicates that the deeper layers represent higher heterogeneity across the noisy and clean clients in Fed-NCL.

The features learned from the noisy clients’ model demonstrate notable distinctions from the ones obtained from the clean clients’ model. Therefore, it is crucial to investigate how the noisy clients affect the global model during training. To achieve this, we compute the CKA similarity between each layer of the global model and the local models of both clean and noisy clients. Subsequently, we average each layer’s CKA similarity between the global model and local models of both clean and noisy clients as

\[
\text{CKA}_{\text{Avg}} = \frac{1}{|M|} \sum_{i=1}^{M} \text{CKA}^i (Z_i, Z_G)
\]

(8)

where \( M \) denotes the model set of clean and noisy clients, and \( Z_i \) and \( Z_G \) represent the feature extracted by layer \( l \) of local model \( i \) and global model \( G \), respectively. By averaging each layer’s feature comparison results, we obtain an approximation of each layer’s feature output similarity between the global model and noisy/clean clients’ models. Fig. 3 shows the experimental results with different training rounds. In Fig. 3(a), we can observed that both the clean and noisy clients have the lower CKA similarity in the deeper layer at the start of training, which indicates that the noisy clients have less impact of global model. As the training progresses, it can be observed from Fig. 3(b) that the noisy clients exhibit significantly lower CKA similarity scores with the global model in deeper layers, when compared with the clean clients. This lower CKA feature similarity score between the global model and noisy clients signifies that the local model from the noisy clients diverges from the global model in terms of feature extraction ability. Despite the presence of noisy clients, the global model’s feature extraction ability still exhibits high consistency with clients with clean labels, reflecting a high CKA similarity score with clean clients. The above experimental study on Fed-NCL provides insight into the impact of noisy labels on feature extraction ability. The study shows that in comparison to training on clients with clean labels, the feature extraction ability of deeper layers in the noisy clients’ model severely declines. This decline is reflected by the lower feature similarity among the noisy and clean clients’ deeper layers. Furthermore, the deeper layer of the noisy clients’ models can
Fig. 4. Workflow of our Fed-NCL. The workflow contains five steps: 1) the server sends the global model $\Theta_G$ to each client; 2) local training on private data; 3) each client sends its local model and loss to the server; 4) the server detects noisy clients by measuring the clients’ reliability scores; and 5) the server performs layerwise aggregation by weight matrix $W_t$. 

be biased by their noisy label data, resulting in divergence from the global model.

Based on the above empirical observations, we can analyze why FedAvg fails to handle noisy clients in FL. In Fed-NCL, client’s local training is typically optimized using stochastic gradient descent (SGD) with backpropagation. It is important to note that the supervision signals gradually propagate through the network from the later layers (closer to the output layer) to the earlier layers (closer to the input layer). Since the optimization procedure follows the empirical risk after the output layer, the impact of noisy labels may be more severe for the later layers. This is because the later layers have already processed more information and are more sensitive to the changes in the input data. As a result, the impact of noisy labels on the later layers can be more significant. It suggests that the influence of noisy labels differs depending on the layer within the model. Therefore, in terms of optimizing the global model, FedAvg [10] depends on combining the gradients or weights provided by all the clients to adjust the shared model. However, if a client has noisy gradients or weights due to incorrect or biased data, it can hinder the optimization process for the entire model. The noisy contributions from these clients can negatively influence the layer optimization goal of the global model.

In summary, the above experimental results have become apparent that the current workflow is not adequately equipped to handle the presence of noisy clients. The deleterious effects of such clients on model convergence, convergence rate, accuracy, and feature extraction ability have been well-documented. Unfortunately, the ubiquity of noisy clients in real-world settings means that this issue cannot be ignored. Given the critical role that FL plays in many applications, it is imperative that a robust framework be developed to address the challenges posed by noisy clients. This framework should be designed with the specific needs of real-world scenarios in mind and should prioritize the optimization of model performance in the presence of such clients. Only by tackling this issue head-on can we ensure the continued viability and success of FL in practice.

V. FEDERATED NOISY CLIENT LEARNING

Based on the empirical results in Section IV, directly aggregating the local models trained from noisy clients can harm the performance of the global model. Therefore, it is crucial to identify noisy clients and mitigate their negative effects. Dealing with noisy clients in FL presents two main technical challenges.

1) How to efficiently identify clients with heterogeneous noise levels while preserving data privacy?
2) How to conduct a robust model aggregation to effectively mitigate the impact of noisy clients?

To tackle the above challenges, we introduce Fed-NCL, which effectively distinguishes noisy clients and intelligently mitigates their impact during the overall FL process. An overview of Fed-NCL, as shown in Fig. 4, mainly contains the following three stages: 1) noisy client detection (Section V-A); 2) robust layerwise adaptation aggregation (Section V-B); and 3) label correction (Section V-C). To identify noisy clients, the server calculates the reliability scores of the clients to statistically determine the noisy clients. After detecting the noisy clients, the server performs robust layerwise adaptation aggregation, which jointly considers the model’s layer divergence and the impact of noisy clients, to obtain a global model for the next round of local training. Finally, we correct the labels from the noisy clients to reduce their negative impact and extract more valuable features from them.

A. Noisy Client Detection.

Identifying noisy clients is crucial to prevent the global model from being negatively impacted by them during the learning procedure. However, the existing methods for noisy client detection suffer from practical limitations and lack universality. For instance, the method proposed by Xu et al. [24]
requires all the clients to participate in the pretraining stage for noisy client detection. This requirement is impractical in cross-device FL as not all the clients have suitable conditions, such as a good network connection and high computational resources, to complete the training process. Moreover, other methods like those proposed by Chen et al. [19] and Tuor et al. [20] rely on benchmark data, which is a set of data with completely clean labels stored on the server, for detecting noisy clients. However, obtaining such clean label data can be challenging in real-world applications. To address these challenges, we propose a universal and benchmark data-free approach for noisy clients’ detection. Our approach involves assigning a reliability score to each participant through jointly considering the quality of their data and the divergence between their local model and the global model. Subsequently, the server uses an abnormal detection model to identify noisy clients by analyzing the distribution of the reliability scores across all the participants. Importantly, our approach neither relies on benchmark data nor forces all the clients to join the training, making it more practical and efficient to implement in real-world scenarios.

1) Reliability Score: To effectively identify noisy clients, we introduce the concept of reliability score to assess the clients. Considering the substantial dynamism and heterogeneous noise levels observed among the participating clients during each training round [22], [33], it becomes crucial for the reliability score to dynamically adapt to accurately capture the varying noise levels exhibited by the clients. Inspired by the observations in Section IV, which highlight the significant variability in feature extraction from the global model when trained on noisy clients, the reliability score first quantifies the discrepancy between the local and global models to dynamically capture noisy clients. To compute the model divergence between the global model and client $c$ at training round $t$, we use the Euclidean distance between the sets of parameters of the aggregated global model and the local model on client $c$

$$e^c_t = ||\Theta^G_t - \Theta^c_t||^2$$

where $\Theta^G_t$ represents the set of parameters of the aggregated global model with model averaging [10] in the current training round $t$, while $\Theta^c_t$ is the set of parameters of the local model on client $c$. The divergence between the global and local models is illustrated in Fig. 5. As the training proceeds, the local model of a clean client aligns closely with the updated global model, resulting in a small distance between them. However, the distance between the local model of a noisy client and that of a clean client remains substantially large. Thus, model discrepancy is an effective metric to quantify the dynamics [10], [11] in FL.

To address the challenges posed by clients with varying levels of noise, the reliability score is used to assess the data quality of each client through the use of aggregated CE loss. Empirical evidence [9], [17], [34], [35], [36] in deep learning has demonstrated that the evolution of the model’s representation space can be divided into two distinct phases. First, there is an initial phase of dimensionality compression, during which the model tends to capture the underlying true data distribution. Subsequently, there is a later phase of dimensionality expansion, where the model starts to overfit to the presence of noisy labels. Drawing on this evidence, DNNs have demonstrated the ability to effectively differentiate between clean and noisy label data, enabling them to make confident predictions on clean label data. Consequently, clients with clean label data are expected to exhibit a low empirical risk. We use the local training data on the collaborative model to calculate aggregated CE loss

$$h^c_t = \sum_{(x_i, y_i) \in D^c} CE(f(x_i; \Theta^c_t), y_i).$$

By integrating the model divergence (9) and data quality metrics (10), we define the reliability score for each client, which is used to identify and mitigate the impact of noisy clients during the FL process as follows:

$$q^c_t = \frac{e^c_t \times h^c_t}{|D^c|}$$

where $|D^c|$ is the amount of training data. The reliability score consists of two metrics to quantify the client’s model, which does not rely on extra benchmark data and adapts to the dynamic of the training process. It fully uses the information of the local model while preserving the data privacy of the clients. Fig. 6 presents the distribution of reliability scores across clean and noisy clients in different experimental scenarios, showing a clear margin between the two groups of clients. Hence, the reliability score is an efficient selection criterion to distinguish noisy clients from clean clients.

2) Noisy Clients’ Detection: The detection of noisy clients is a crucial consideration in the context of FL. After obtaining the reliability score of each client, the server can distinguish between noisy and clean clients using the reliability scores $Q_t = \{q^1_t, \ldots, q^n_t\}$ for each client. We approach the identification of noisy clients as an abnormal detection task. While a fixed threshold or an anomaly detection model, such as
an autoencoder, can be used to filter out noisy clients, these methods may not guarantee the detection performance due to the highly dynamic and unpredictable training process in FL.

To address these limitations, we propose using anomaly detection with the normal distribution to identify noisy clients. In this approach, we approximate all the clients’ reliability scores as normally distributed. As per the conventional heuristic rule in statistics, nearly all the values (99.7%) in a normal distribution lie within 3 standard deviations $\sigma$ of the mean. Therefore, if a client $c$ violates this rule, it will be identified as a noisy client

$$q_i^c = E(Q_i) > \beta \sigma(Q_i)$$

where $\beta$ represents the number of standard deviations of the mean. By adjusting $\beta$, we can ensure a high accuracy of anomaly detection that can adapt to the high dynamics of FL training. Upon applying this procedure, we divide the set of clients $S$ into two subsets: $S_n$ (noisy clients) and $S_c$ (clean clients). This approach enables us to accurately identify noisy clients in FL, thereby enhancing its overall efficiency and effectiveness.

### B. Noise Robust Layerwise Aggregation.

To mitigate the negative impact caused by noisy clients, we introduce a noise-robust layerwise aggregation technique. Building upon the insights from Section IV, we observe a significant deterioration in the feature extraction capability of deeper layers in the models of noisy clients. This observation highlights the varying impact of noisy clients on different layers of the model. Unfortunately, the existing methods [19], [20], [21], [24] for handling noisy clients in FL fail to consider this distinction and instead rely on the conventional FedAvg approach [10] to equally aggregate all the local models. This leads to suboptimal performance of the global model when dealing with noisy clients. Our noise-robust layerwise aggregation technique addresses this limitation by adapting the aggregation process to the varying impact of noisy clients on different layers.

Specifically, our noise-robust layerwise aggregation introduces a weighting scheme that assigns higher weights to the models from clients with lower noise ratio in the deeper layers, while giving more weight to the models from clients with high noise ratio in the shallower layers. We apply an aggregation weight matrix $W_c$ at the server to reduce the negative effect from noisy clients in terms of layer

$$W_c = [w_1^c, w_2^c, \ldots, w_l^c]$$

where $w_l^c$ represents the aggregation weight vector of the $l$th layer at round $t$, while $w_l^c$ represents the aggregation weight for client $c$ in the $l$th layer. For each layer $l$, $\sum_{c=1}^C w_l^c = 1$. For each client $c$, we quantify the distance $d_l$ with the global model of each layer at round $t$ as follows:

$$d_l^c = 1 + ||\Theta_l^{G} - \Theta_l^c||^2, l \in \{1, \ldots, L\}$$

where $L$ is the number of layers, and $l$ is the layer index of the model. According to the training data distribution on different clients, for each layer $l$, the distance $D_l^t$ to the global model at round $t$ is defined as

$$D_l^t = \frac{\sum_{c=1}^C w_l^c D_{l}^c}{\sum_{c=1}^C w_l^c}.$$  \hspace{1cm} (15)

It is intuitive to discard models from noisy clients during global aggregation because they disrupt the optimization of the global model. However, this approach slows down the convergence process of the global model since it only learns from clients with clean label data, resulting in poor generalization. Based on the aforementioned observation, the high similarity in output features of the previous layer between local models from noisy and clean clients suggests that noisy clients can still learn the same patterns as clean clients, making their models valuable for the global model. To address this, we introduce a parameter, denoted as $\tau$, to increase the distance between the global model and the models from noisy clients while mitigating their negative impact and improving the generalization of the global model. Despite the presence of noisy clients, they can initially learn clean and simple patterns due to the memorization effect of DNNs. The issue arises when the local models from noisy clients gradually overfit their noisy label data, thereby compromising the global model. To tackle this problem, we propose an adaptive penalty control mechanism, denoted as $m(c)$, specifically designed for noisy clients

$$m(c, T) = \min\left(\frac{T}{T_k}, \tau \right) c \in S_n, 1 c \in S_c.$$  \hspace{1cm} (16)

Specifically, to ensure the boundedness of the penalty factor as the value of $T$ increases, it is essential for the function $m(c)$ to exhibit a nonincreasing behavior with respect to $T$. This restriction ensures that the penalty factor does not become unbounded as $T$ increases. The specific value of $T_k$ depends on the noise rate present in the system. Then, for each layer $l$, we compute a weighted sum to mitigate the negative impact from the noisy clients while improving the generalization of the global model as

$$w_l^c = \frac{m(c, T) D_l^c}{\sum_{c=1}^C m(c, T) D_l^c}.$$  \hspace{1cm} (17)

Subsequently, we perform layerwise weighted aggregation of the local models and update the global model as follows:

$$\Theta_{i+1}^l \leftarrow \sum_{l=1}^L \sum_{c=1}^C w_l^c \Theta_l^c.$$  \hspace{1cm} (18)

where $\Theta_{i+1}^l$ represents the parameters of the aggregated global model, and $\Theta_l^c$ denotes the parameters of layer $l$ received from client $c$ after the $i$th global training round. Notably, these weights are inversely proportional to the distance between the given local model and the updated global model. Consequently, as the training progresses, the global model adjusts its parameters in accordance with the guidance provided by the clean clients, thereby moving in the direction guided by them.
C. Label Correction

To fully use the local data, we perform the label correction on the noisy clients’ local data. We follow the two stages of training procedure like [37]. Suppose the global model converges after $T_{\text{corr}}$ round training, we apply the label correction on the noisy clients $S_c$. To ensure the label correction performs on the real noisy clients, we extend the validation time interval at different training iterations to identify the real noisy clients. At each global training round, the clients are divided into noisy clients and clean clients based on the method we proposed in Section V-A. Before $T_{\text{corr}}$-1 round, the server stores each round’s results of noisy client detection. The noisy clients $S_{\text{corr}}$ that need to conduct label corrections are determined as

$$S_{\text{corr}} = \left( \frac{\sum_{i=1}^{\tau_c} |c \in S^c_i|}{\sum_{i=1}^{\tau_c} \epsilon} \right) > \alpha T_{\text{corr}}. \quad (19)$$

Then, each noisy client $c \in S_{\text{corr}}$ performs the label correction of all the local data using the predicted label from the global model $\Theta_G$ to generate a new pseudolabel. To avoid overcorrection, we only relabel the instances with confidence exceeding the threshold $\eta$. Thus, for each noisy client $c \in S_{\text{corr}}$, the subset $D_c$ of relabel samples is given by

$$D_c = \{ (x^c, ̂y^c) | ̂y^c = \max \{ f(x^c; \Theta_G^c) > \eta \} \}. \quad (20)$$

After performing the label correction on the noisy clients $S_{\text{corr}}$, the new relabel subset $D$ is used for local model training.

VI. EXPERIMENTS

A. Experimental Setup

1) Datasets and Models: We conduct experiments based on four datasets: 1) MNIST [38]; 2) FASHIONMNIST [39]; 3) CIFAR10 [29]; and 4) CIFAR10N [8] which are widely used for the evaluation of noisy labels in previous works [17]. In our experiment, we evaluate the performance of Fed-NCL on the following widely used deep learning models: Lenet5 [40], VGG16 [30], and vision transformers (ViT(S)) [41].

2) Baseline: The efficacy of Fed-NCL is appraised against three FL baselines. FedAvg [10] performs local training and aggregates on a central server, weighting by the client’s data volume. Trimmed mean [42] independently processes each model parameter. The server organizes, trims extremes, and calculates the mean of the remaining parameters for the global model. FedProx [11] adds a proximal term with $\mu = 0.01$ in the local optimization objective to counter local data heterogeneity issues, ensuring closer alignment to the global model. FOCUS [19] uses benchmark samples to assess and adjust participant weights based on local data credibility, determined by computing mutual CE between models.

3) Noise Distribution: In the experiment, we evaluate the performance of Fed-NCL under different noisy data distributions. Specifically, the following two common distributions are adopted to characterize the noise scenario across different clients: 1) Bernoulli distribution and 2) truncated Gaussian distribution. For the label noise injection, we use the symmetric noise approach as outlined in Song et al. [13]. This approach involves manually injecting noise into the dataset through the use of a label transition matrix $T$, where $T_{ij} := p(\tilde{y} = j | y = i)$. This matrix describes the probability of a true label $i$ being corrupted and flipped into a different label $j$. Specifically, we use an equal probability distribution for flipping true labels into other labels, resulting in a symmetric noise approach.

4) Data Distribution: For realistic simulation in our FL framework, both the IID and non-IID data partitions are considered. IID data are evenly distributed among C clients. For non-IID partitions, class and quantity skewness are simulated. Class skewness follows Xu’s approach [24], using the Bernoulli and symmetric Dirichlet distributions to create diverse class distributions among clients. Quantity skewness is mirrored by sampling training data sizes from a lognormal distribution, standardizing the sample size deviation at 0.3, thereby effectively representing variability in training data volumes across clients.

5) Implementation Detail: In this study, all the experiments involving convolutional neural network (CNN) architectures used a consistent setup. Specifically, the number of participating clients was set to $N = 20$, with a total of 150 global rounds and $T_{\text{corr}} = 60$. SGD was implemented as the local training optimizer with a learning rate of 0.01, a local batch size of 128, and local training epoch $E$ of 10 for all the datasets. Our method, Fed-NCL, was implemented with the hyperparameters $\alpha = 0.6$ and $\tau = 50$ across all the experiments. In scenarios involving Bernoulli or Gaussian noise distribution and all the datasets, we set $\beta$ to 0.7 for all the experiments. For experiments involving transformers, ViT(S) [41] were used, and the experiments were conducted over 100 communication rounds and local training epoch $E$ of one round. To ensure stable training, a linear learning rate with 500 epoch warm-up and decay scheduler was used for our ViT-FL model. In addition, gradient clipping with a global norm of 1 was applied to enhance stability. Our experiments with transformers were conducted with hyperparameters $\alpha = 0.6$ and $\tau = 1$, consistent with our other experiments.

B. Evaluation With Heterogeneous Noise Distributions

In this section, we first evaluate the effectiveness of our Fed-NCL on two noise scenarios proposed in Section III, when the training data are uniformly distributed among the participating clients. Table I shows the evaluation results of different schemes when the noisy data follow the Bernoulli distribution and truncated Gaussian distribution. The experimental results demonstrate that Fed-NCL outperforms the other baselines in various noise distributions. For Bernoulli distribution, the Trimmed mean and FedProx cannot address the noise clients’ problem as their average test accuracy drops significantly when increasing the number of noisy clients, by 24.23% for the Trimmed mean and 12.16% for the FedProx, respectively. For Fed-NCL, it only drops 2.34%. Significantly, Fed-NCL outperforms other baselines by at least 11.23% in MNIST, 19.87% in Fashion MNIST, and 29.37% in CIFAR10 in the S1 setting (60% of clients with noisy labels data). These results point out that Fed-NCL is robust to noisy clients, especially in the high ratio of noisy clients. For the truncated Gaussian distribution, the corresponding results show that
Fed-NCL also can effectively mitigate the impact of noisy clients which follows a truncated Gaussian distribution, which improves the FedAvg at least by 7.96% in MNIST, 7.44% in Fashion MNIST, and 9.49% in CIFAR10. Figs. 7 and 8 show the global model’s test accuracy of different schemes when the noisy data follow a Bernoulli distribution with $p = 0.7$ across different clients and truncated Gaussian distribution with $(\mu = 0.4, \sigma = 0.45)$, respectively. Specifically, for the Bernoulli distribution, we can see that the global model trained by FedAvg and FedProx cannot converge because their test accuracy curves are unstable at the end of training. This is because the noisy clients guide the update of the collaborative model in a divergent direction during the model aggregation process. Although the FedProx adds the extra proximal term to reduce the variance of local updates, it cannot effectively mitigate the negative effect from the noisy data. The Trimmed mean performs better by removing a specific percentage of the smallest and largest parameters from the collected local models. However, Fed-NCL achieves the prominent performance advantage among all the baselines for the following reasons. During the training process, Fed-NCL can prevent the misleading from the noisy clients’ model by correctly identifying the noisy clients from clean clients. The reliability score can effectively evaluate each client by measuring the quality of the local training data and model. Thus, correctly detecting the noisy clients is the first key success of Fed-NCL. Then, our robust layerwise aggregation can reduce the impact of the noisy clients and guide the global model’s updates with clean clients’ model, increasing the model convergence rate. Finally, Fed-NCL dynamically performs label correction on the noisy clients, which improves the generalization ability of the global model.

C. Evaluation With Heterogeneous Data Distributions

This section evaluates the performance of various models, focusing on scenarios involving participating clients with heterogeneous local training data and noisy label data. Table II details the performance in situations where noisy data conform to Bernoulli and truncated Gaussian distributions with

| TABLE I |
|---------------------------------|
| ACCURACY OF DIFFERENT SCHEMES IN VARIOUS SCENARIOS WHEN EACH CLIENT HAS THE SAME AMOUNT OF LOCAL TRAINING DATA. WE REPORT THE AVERAGE ACCURACY OVER THE LAST TEN ROUNDS. (S1-BERNULLI DISTRIBUTION WITH p = 0.6; S2-BERNOULLI DISTRIBUTION WITH p = 0.7; S3-BERNOULLI DISTRIBUTION WITH p = 0.8; S4-TRUNCATED GAUSSIAN DISTRIBUTION WITH (μ = 0.4, σ = 0.45); S5-TRUNCATED GAUSSIAN DISTRIBUTION WITH (μ = 0.3, σ = 0.45); S6-TRUNCATED GAUSSIAN DISTRIBUTION WITH (μ = 0.3, σ = 0.4)). * REPRESENTS THE PERFORMANCE IS OMITTED SINCE THE PERFORMANCE IS POOR. |
|---------------------------------|
|---------------------------------|
| Lenet5 on MNIST                 |
|---------------------------------|
| Ours                             | 98.87% | 98.84% | 99.12% | 97.24% | 97.65% | 97.04% |
| FedAvg                          | 87.64% | 93.96% | 88.00% | 86.63% | 88.78% | 89.08% |
| Trimm                           | 82.62% | 97.59% | 98.72% | 91.35% | 95.01% | 94.46% |
| FedProx                         | 83.24% | 88.27% | 96.12% | 85.73% | 88.90% | 89.79% |
| FOCUS                           | -      | 77.28% | 96.55% | 55.58% | 62.90% | 55.38% |
|---------------------------------|
| Lenet5 on Fashion MNIST        |
|---------------------------------|
| Ours                             | 87.48% | 88.53% | 88.85% | 85.29% | 86.23% | 86.73% |
| FedAvg                          | 67.61% | 81.30% | 75.93% | 75.23% | 78.32% | 79.29% |
| Trimm                           | 60.55% | 76.55% | 85.95% | 78.14% | 82.54% | 81.79% |
| FedProx                         | 65.91% | 76.29% | 76.60% | 79.02% | 81.23% | 81.22% |
| FOCUS                           | -      | 56.40% | 64.71% | 50.97% | 54.47% | 61.01% |
|---------------------------------|
| VGG16 on CIFAR10                |
|---------------------------------|
| Ours                             | 74.82% | 77.97% | 78.28% | 59.66% | 67.37% | 63.39% |
| FedAvg                          | 42.82% | 51.16% | 58.19% | 51.17% | 54.99% | 57.01% |
| Trimm                           | 41.85% | 64.87% | 73.09% | 54.25% | 62.45% | 60.82% |
| FedProx                         | 43.50% | 48.84% | 56.41% | 50.65% | 55.88% | 55.95% |
| FOCUS                           | -      | 58.97% | 56.17% | -      | 26.95% | 30.64% |
unbalanced local training data across clients. Here, Fed-NCL consistently excels, outperforming other methods by up to 35%. In conditions of heterogeneous data distribution under similar noise scenarios, as detailed in Table III, Fed-NCL again leads, outpacing other methods by up to 26%. Despite all the baseline methods witnessing a performance drop compared with the IID setting, Fed-NCL maintains its effectiveness across varied noise scenarios and data heterogeneity. To counteract the challenges posed by simultaneous data heterogeneity and noisy clients, Fed-NCL distinguishes noisy clients by analyzing data quality and model divergence. This identification allows for precise global model updates using SGD, followed by a robust layerwise aggregation, resulting in a reinforced global model. This approach confirms the efficiency of Fed-NCL in managing noisy clients amidst data heterogeneity, reinforcing its comprehensive applicability in such environments.

### D. Ablation Study

In this section, to study the effectiveness of Fed-NCL, we conduct the ablation study of Fed-NCL on the CIFAR-10 in two noise scenarios proposed in Section III with different data partitions. Table IV shows the corresponding result. The result shows that adding the penalty for noisy clients in layerwise model aggregation has a critical contribution to Fed-NCL. Without the penalty for noisy clients’ local models, the performance of global models can decrease at most 26.58% and 25.32% in IID and non-IID data partitions, respectively. It represents that adding the penalty for the noisy clients in layerwise model aggregation can prevent the negative effect from the noisy clients during the model aggregation. Thus, it can improve the effectiveness and robustness of global training in Fed-NCL. For the label correction, it can improve the global model performance up to 13% in serious noise settings (30% of clients have data with noisy labels). As for the noisy client’s detection, precisely measuring the reliability score of each client is the key to success in effectively detecting noisy clients. Thus, we removed two corresponding criteria in (9) and (10) in reliability score, respectively. We can observe that both the criteria contribute to reliability score, which improves global model’s performance 6% by on average.

To perform a comprehensive sensitivity analysis for the hyperparameters in Fed-NCL, we conducted a series of experiments on the CIFAR-10 dataset using two distinct noise scenarios as described in Section III. Our objective was to evaluate the impact of three key hyperparameters, namely, the noisy clients detection std $\beta$, label correction confident threshold $\eta$, and label correction time factor $\alpha$. Our analysis began by studying the interaction between the noisy clients detection std $\beta$ and model performance. As shown in Fig. 9(a) and 9(b), we observed that the lowest accuracy was obtained when $\beta = 0.7$. However, increasing the value of $\beta$ to 1.0 did not lead to a decline in performance under the Bernoulli noise scenario. Notably, $\beta$ is responsible for controlling the sensitivity of the noisy clients, and decreasing it may lead to a reduction in the accuracy of noisy clients’

### TABLE II

| Dataset/Model | Accuracy on MNIST | Accuracy onFashion-MNIST | VGG16 on CIFAR10 |
|---------------|-------------------|--------------------------|------------------|
| Ours          | 98.72%            | 84.34%                   | 63.17%           |
| FedAvg        | 82.75%            | 73.85%                   | 51.47%           |
| Trinn         | 63.32%            | 63.22%                   | 47.35%           |
| FedProx       | 83.51%            | 76.03%                   | 65.94%           |
| FOCUS         | -                 | 97.27%                   | 67.38%           |

### TABLE III

| Dataset/Model | Accuracy on MNIST | Accuracy on Fashion-MNIST | VGG16 on CIFAR10 |
|---------------|-------------------|--------------------------|------------------|
| Ours          | 98.79%            | 94.15%                   | 63.17%           |
| FedAvg        | 94.23%            | 85.43%                   | 51.47%           |
| Trinn         | 77.81%            | 76.84%                   | 51.47%           |
| FedProx       | 90.64%            | 88.95%                   | 51.47%           |
| FOCUS         | 77.91%            | 42.79%                   | 51.47%           |

### TABLE IV

| Data partition | ID   | Non-ID |
|----------------|------|--------|
| Noise Setting  | S1   | S2     | D1   | D2   |
| w/o label correction | 84.84% | 76.31% | 55.26% | 50.6% |
| w/o penalty     | 81.12% | 74.60% | 47.90% | 54.40% |
| w/o criterion (9) | 74.27% | 70.03% | 56.98% | 59.34% |
| w/o criterion (10) | 70.39% | 73.15% | 48.64% | 54.61% |
| Ours            | 77.97% | 78.29% | 58.65% | 65.24% | 77.19% | 78.59% | 57.18% | 61.90% |
that can be used in various practical settings. Highlighting the potential of Fed-NCL as a reliable FL algorithm for participating clients. Therefore, the results of the ablation study show that Fed-NCL is a highly effective and robust approach to real-world FL scenarios, regardless of the number of participating clients. This indicates that the performance does not exhibit a significant decline (<10%) as the number of participating clients increases. This suggests that a higher threshold value of 0.7, A\_\text{REPRESENTED}\_\text{USING}\_\text{THE}\_\text{BERNOULLI}\_\text{DISTRIBUTION}\_\text{CHALLENGED}\_\text{BY}\_\text{A}\_\text{MEAN}\_\text{(|\mu|)\_OF}\_\text{0.3}\_\text{AND}\_\text{A}\_\text{STANDARD}\_\text{DEVATION}\_\text{(|\sigma|)\_OF}\_\text{0.4}, \_\text{REPRESENTS}\_\text{THE}\_\text{GAUSSIAN}\_\text{DISTRIBUTION}. \_\text{THE}\_\text{AVERAGE}\_\text{ACCURACY}\_\text{OF}\_\text{THE}\_\text{LAST}\_\text{TEN}\_\text{ROUNDS}\_\text{IS}\_\text{REPORTED}.}

| Bernoulli Distribution | 20 | 50 | 80 | 100 |
|------------------------|----|----|----|-----|
| Bernoulli Distribution | 77.57% | 74.15% | 70.18% | 69.61% |

### B. Communication and Computation Efficiency
Fed-NCL demonstrates remarkable efficiency in both communication and computation for the client in cross-device FL [2]. The detection of noisy clients in Fed-NCL showcases its prowess in communication efficiency. In contrast to the traditional FedAvg process, Fed-NCL only requires the participation of the noisy clients' CE values to the server. This aggregated CE, being a numerical value, is negligible in comparison to the model communication overhead. Indeed, the communication efficiency of Fed-NCL surpasses that of the existing methods [19], [20], [21]. Furthermore, Fed-NCL imposes no additional computation burden on the client during the detection of noisy clients. In contrast, other methods such as FedCorr [24], FOCUS [19], RoFL [21], and DS [20] require additional model inference on the client's entire training data prior to local model training in each communication round.

### C. Comparing Fed-NCL With Previous Approaches to Federated Noisy Learning
First, Fed-NCL is designed to address the complexities of real-world FL scenarios involving clients with heterogeneous noise levels. In contrast to the existing studies, which often make impractical assumptions such as homogenous noise levels among all the clients [21] or the exclusive presence of clean data [19], [20], [24], Fed-NCL introduces two types of realistic noisy scenarios. By precisely modeling noisy clients, Fed-NCL enables adaptable and universal validation techniques, serving as an evaluative mechanism for real-world FL methodologies.

Second, the prevailing focus in FL research is the improvement of global model performance, often overlooking the nuanced impacts of noisy clients on the model's individual layers. Fed-NCL fills this knowledge gap by conducting the first systematic empirical study that quantifies the negative impact of noisy clients on diverse layers of the global model. These findings catalyze future FL research, promoting the development of more efficient and robust techniques.

### Table V

| Bernoulli Noise Levels, With a Probability of Success (p) of 0.7, Are Represented Using the Bernoulli Distribution. The Truncated Gaussian Distribution, Characterized by a Mean (\mu) of 0.3 and a Standard Deviation (\sigma) of 0.4, Represents the Gaussian Distribution. The Accuracies of the Last Ten Rounds Are Reported. |
|-----------------|-----|-----|-----|-----|
| N               | 20  | 50  | 80  | 100 |
| Bernoulli       | 77.57% | 74.15% | 70.18% | 69.61% |
| Gaussian        | 63.39% | 61.04% | 58.96% | 59.89% |

**Fig. 9.** Ablation study of Fed-NCL with different noisy client distributions. Bernoulli distribution with p = 0.7. Gaussian distribution: \mu = 0.3, \sigma = 0.4. (a) Bernoulli distribution. (b) Gaussian distribution. (c) Bernoulli distribution. (d) Gaussian distribution. (e) Bernoulli distribution. (f) Gaussian distribution.

**Fig. 9(e) and 9(f),** our results revealed that changing the label correction threshold factor \( \alpha \) can result in more high-quality pseudolabels during label correction. Finally, we analyzed the impact of the label correction time factor \( \beta \) on model performance. As presented in Fig. 9(c) and 9(d), we observed that a lower threshold value of 0.7 yields the best performance in both the noise scenarios. Although increasing the threshold to 0.8 or 0.9 did not result in a significant decrease in performance, a small threshold value led to a reduction in accuracy by more than 1.5%. This suggests that a higher threshold value can result in more high-quality pseudolabels during label correction.

**Table V**

### VII. Discussion

#### A. Limitation and Scalability
This article primarily focuses on modeling and discussing the distribution of noise levels across a large number of clients exhibiting varying degrees of noise. We acknowledge the existence of different types of label noise [17], [43], including label random flipping, instance-dependent noise, and systematic noise. We recognize the importance of understanding and addressing different types of label noise, but our research scope primarily centers around evaluating the impact of noise on the performance and robustness of FL across various scenarios. Incorporating insights from our article regarding the effects of different noise types on the performance and reliability of FL can inform the design of more targeted methods in the future.

#### B. Communication and Computation Efficiency
Fed-NCL demonstrates remarkable efficiency in both communication and computation for the client in cross-device FL [2]. The detection of noisy clients in Fed-NCL showcases its prowess in communication efficiency. In contrast to the traditional FedAvg process, Fed-NCL only requires the participation of noisy clients to transmit their aggregated CE values to the server. This aggregated CE, being a numerical value, is negligible in comparison to the model communication overhead. Indeed, the communication efficiency of Fed-NCL surpasses that of the existing methods [19], [20], [21]. Furthermore, Fed-NCL imposes no additional computation burden on the client during the detection of noisy clients. In contrast, other methods such as FedCorr [24], FOCUS [19], RoFL [21], and DS [20] require additional model inference on the client’s entire training data prior to local model training in each communication round.

#### C. Comparing Fed-NCL With Previous Approaches to Federated Noisy Learning

First, Fed-NCL is designed to address the complexities of real-world FL scenarios involving clients with heterogeneous noise levels. In contrast to the existing studies, which often make impractical assumptions such as homogenous noise levels among all the clients [21] or the exclusive presence of clean data [19], [20], [24], Fed-NCL introduces two types of realistic noisy scenarios. By precisely modeling noisy clients, Fed-NCL enables adaptable and universal validation techniques, serving as an evaluative mechanism for real-world FL methodologies.

Second, the prevailing focus in FL research is the improvement of global model performance, often overlooking the nuanced impacts of noisy clients on the model’s individual layers. Fed-NCL fills this knowledge gap by conducting the first systematic empirical study that quantifies the negative impact of noisy clients on diverse layers of the global model. These findings catalyze future FL research, promoting the development of more efficient and robust techniques.
Moreover, Fed-NCL distinguishes itself as an innovative, efficient, and privacy-preserving approach. Unlike existing methods that merely adapt centralized learning techniques to the federated context [20], [21], [24], Fed-NCL tries to solve this problem from a novel perspective. For example, methods such as FedCorr [24] and DS [20] introduce computational and communicative overheads. Specifically, FedCorr necessitates an additional inference round to identify noisy clients, while DS requires each client to transmit individual sample losses for benchmark comparison. In contrast, Fed-NCL uses a universal, effective method for noisy client detection. It introduces a reliability score designed on the basis of empirical observations in Fed-NCL, allowing for precise data quality assessment without assumptions. In addition, Fed-NCL designs layerwise aggregation to more finely attenuate the adverse effects of noisy clients, thereby facilitating stable global model convergence while maintaining performance.

VIII. Conclusion

In this article, we first investigate the critical issue caused by noisy clients in FL and quantify the negative impact of the noisy clients in terms of the representations learned by different layers. We have shown that the noisy client could negatively impact the convergence, feature extraction ability, and performance of the aggregated global model in FL. Moreover, to effectively conduct FL with noisy clients, we have proposed a simple yet effective learning framework, Fed-NCL (Fed-NCL), which first identifies the noisy clients through well estimating the data quality and model divergence. Then robust layerwise aggregation is proposed to adaptively aggregate the local models of each client to deal with the data heterogeneity caused by the noisy clients. We further perform the label correction on the noisy clients to improve the generalization of the global model. The experimental results have shown that our Fed-NCL can effectively boost the performance of different deep FL with noisy clients in homogeneous and heterogeneous FL.

References

[1] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” 2016, arXiv:1610.02557.

[2] P. Kairouz et al., “Advances and open problems in federated learning,” Dec. 2019, arXiv:1912.04977.

[3] C. Niu et al., “Billion-scale federated learning on mobile clients: A submodel design with tunable privacy,” in Proc. 26th Annu. Int. Conf. Mobile Comput. Netw., Sep. 2020, pp. 1–14.

[4] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, “Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results,” Med. Image Anal., vol. 65, Oct. 2020, art. no. 101765.

[5] D. Byrd and A. Polychroniadou, “Differentially private secure multi-party computation for federated learning in financial applications,” 2020, arXiv:2010.05867.

[6] C. Zhang, Z. Yu, H. Fu, P. Zhu, L. Chen, and Q. Hu, “Hybrid noise-oriented multilabel learning,” IEEE Trans. Cybern., vol. 50, no. 6, pp. 2837–2850, Jun. 2020.

[7] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 2691–2699.

[8] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu, “Learning with noisy labels revisited: A study using real-world human annotations,” 2021, arXiv:2110.12088.

[9] D. Arpit et al., “A closer look at memorization in deep networks,” in Proc. ICML, 2017, pp. 233–242.

[10] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Proc. Artif. Intell. Statist., vol. 54, Feb. 2017, pp. 1273–1282.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” 2018, arXiv:1812.06127.

[12] B. Frey and M. Verleysen, “Classification in the presence of label noise: A survey,” IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 5, pp. 845–869, May 2014.

[13] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning from noisy labels with deep neural networks: A survey,” 2020, arXiv:2007.08199.

[14] A. Menon, B. V. Rooyen, C. S. Ong, and B. Williamson, “Learning from corrupted binary labels via class-probability estimation,” in Proc. ICML, 2015, pp. 125–134.

[15] E. Malach and S. Shalev-Shwartz, “Decoupling ‘when to update’ from ‘how to update,’” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 960–970.

[16] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, “MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels,” in Proc. Int. Conf. Mach. Learn., vol. 80, 2018, pp. 2304–2313.

[17] B. Han et al., “Co-teaching: Robust training of deep neural networks with extremely noisy labels,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 8527–8537.

[18] J. Goldberger and E. Ben-Reuven, “Training deep neural-networks using a noise adaptation layer,” in Proc. ICLR, 2017, pp. 1–9.

[19] Y. Chen, X. Yang, X. Qin, H. Yu, B. Chen, and Z. Shen, “FOCUS: Dealing with label quality disparity in federated learning,” 2020, arXiv:2001.11359.

[20] T. Tuor, S. Wang, B. J. Ko, C. Liu, and K. K. Leung, “Overcoming noisy and irrelevant data in federated learning,” in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 5020–5027.

[21] S. Yang, H. Park, J. Byun, and C. Kim, “Robust federated learning with noisy labels,” IEEE Intell. Syst., vol. 37, no. 2, pp. 35–43, Mar. 2022.

[22] K. Tam, L. Li, B. Han, C. Xu, and H. Fu, “Federated noisy client learning,” 2021, arXiv:2106.13259.

[23] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, “Similarity of neural network representations revisited,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 3519–3529.

[24] J. Xu, Z. Chen, T. Q. S. Quek, and K. E. Chong, “FedCorr: Multi-stage federated learning for label noise correction,” 2022, arXiv:2204.04677.

[25] D. Ortego, E. Arazo, P. Albert, N. E. O’Connor, and K. McGuinness, “Towards robust learning with different label noise distributions,” 2019, arXiv:1912.08741.

[26] B. Han et al., “Masking: A new perspective of noisy supervision,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 5836–5846.

[27] J. Mukundhammar, J. Widén, P. Grahn, and J. Rydén, “A Bernoulli distribution model for plug-in electric vehicle charging based on time-use data for driving patterns,” in Proc. IEEE Int. Electric Vehicle Conf. (IEVC), Dec. 2014, pp. 1–7.

[28] G. Hosaoya and H. Yoshima, “Constellation shaping for non-uniform signals based on truncated Gaussian distribution,” in Proc. Int. Symp. Inf. Theory Appl. (ISITA), Oct. 2016, pp. 557–561.

[29] A. Krizhevsky et al., “Learning multiple layers of features from tiny images,” Toronto, ON, Canada, 2009.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. ICLR, Sep. 2014, pp. 1–14.

[31] T. Nguyen, M. Raghu, and S. Kornblith, “Do wide and deep networks learn the same things? Uncovering how neural network representations vary with width and depth,” 2020, arXiv:2010.15527.

[32] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, “No fear of heterogeneity: Classifier calibration for federated learning with non-IID data,” in Proc. Adv. Neural Inf. Process. Syst., vol. 34, 2021, pp. 5972–5984.

[33] H. Yu, S. Yang, and S. Zhu, “Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning,” in Proc. AAAI, Jul. 2019, pp. 5693–5700.

[34] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, “Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results,” Med. Image Anal., vol. 65, Oct. 2020, art. no. 101765.

[35] D. Arpit et al., “A closer look at memorization in deep networks,” in Proc. ICML, 2017, pp. 233–242.

[36] H. B. McMahan, E. Moore, D. Ramage, and B. A. Y. Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Proc. Artif. Intell. Statist., vol. 54, Feb. 2017, pp. 1273–1282.

[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” 2018, arXiv:1812.06127.
[36] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” Commun. ACM, vol. 64, no. 3, pp. 107–115, Mar. 2021.

[37] J. Huang, L. Qu, R. Jia, and B. Zhao, “O2U-Net: A simple noisy label detection approach for deep neural networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3325–3333.

[38] Y. Lecun, “The MNIST database of handwritten digits,” 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[39] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017, arXiv:1708.07747.

[40] Y. L. Cun et al., Handwritten Digit Recognition with a Back-Propagation Network. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1990, pp. 396–404.

[41] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” 2020, arXiv:2010.11929.

[42] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in Proc. Int. Conf. Mach. Learn., 2018, pp. 5650–5659.

[43] H. Cheng, Z. Zhu, X. Li, Y. Gong, X. Sun, and Y. Liu, “Learning with instance-dependent label noise: A sample sieve approach,” 2020, arXiv:2010.02347.

Kahou Tam received the M.S. degree in computer science from the University of Macau, Macau, China, in 2023, where he is currently pursuing the Ph.D. degree. His current research interests focus on federated learning and on-device learning.

Bo Han (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Technology Sydney, Sydney, NSW, Australia, in 2019, advised by Lvor W. Tsang and Ling Chen. He is currently an Assistant Professor of computer science with Hong Kong Baptist University (affiliated with HKBU AI Research Cluster), Hong Kong. He has authored more than 100 technical papers at prominent journals and conferences such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, MLJ, ICML, and NeurlPS. His current research interests include machine learning, deep learning, and artificial intelligence.

Chengzhong Xu (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from Nanjing University, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer engineering from the University of Hong Kong, Hong Kong, in 1993. He is now a Professor of electrical and computer engineering, Wayne State University, Detroit, MI, USA, and the Director of the Cloud Computing Center, Shenzhen Institute of Advanced Technology, Chinese Academy of Science, Beijing, China. His main research interests include networked computing systems, reliability, availability, power efficiency, and security.

Dr. Xu is an Editor of IEEE TRANSACTION ON COMPUTERS, IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTION ON CLOUD COMPUTING, and the Journal of Parallel and Distributed Computing.

Huazhu Fu (Senior Member, IEEE) received the Ph.D. degree from Tianjin University, Tianjin, China, in 2013. He is a Senior Scientist with the Institute of High Performance Computing (HPC), A*STAR, Singapore. Previously, he was a Research Fellow with NTU, Singapore, from 2013 to 2015, a Research Scientist with I2R, A*STAR, from 2015 to 2018, and a Senior Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, from 2018 to 2021. His research encompasses computer vision, AI in healthcare, and trustworthy AI. Dr. Fu serves on the Bio Imaging and Signal Processing Technical Committee (BISP TC) of the IEEE Signal Processing Society. He received the number of awards including Best Paper Award with ICME 2021 and Best Paper Award with MICCAI-OMIA 2022. He is an Associate Editor of IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, and the Journal of Biomedical and Health Informatics.

Li Li received the B.S. degree from Tianjin University, Tianjin, China, in 2011, and the M.S. and Ph.D. degrees in electrical and computer engineering from Ohio State University, Columbus, OH, USA, in 2014 and 2018, respectively. He is currently an Assistant Professor with the University of Macau, Macau, China. His current research interests focus on autonomous driving and distributed learning.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.