DML向 FL+DK的扩展，每个client收集其他client的logits做带权整合

Robust Federated Learning with Noisy and Heterogeneous Clients

Xiuwen Fang[1], Mang Ye[1,2*]
[1]National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
School of Computer Science, Wuhan University, Wuhan, China
[2] Hubei Luojia Laboratory, Wuhan, China
https://github.com/FangXiuwen/Robust_FL

Abstract

Model heterogeneous federated learning is a challenging task since each client independently designs its own model. Due to the annotation difficulty and free-riding participant issue, the local client usually contains unavoidable and varying noises, which cannot be effectively addressed by existing algorithms. This paper starts the first attempt to study a new and challenging robust federated learning problem with noisy and heterogeneous clients. We present a novel solution RHFL (Robust Heterogeneous Federated Learning), which simultaneously handles the label noise and performs federated learning in a single framework. It is featured in three aspects: (1) For the communication between heterogeneous models, we directly align the models feedback by utilizing public data, which does not require additional shared global models for collaboration. (2) For internal label noise, we apply a robust noise-tolerant loss function to reduce the negative effects. (3) For challenging noisy feedback from other participants, we design a novel client confidence re-weighting scheme, which adaptively assigns corresponding weights to each client in the collaborative learning stage. Extensive experiments validate the effectiveness of our approach in reducing the negative effects of different noise rates/types under both model homogeneous and heterogeneous federated learning settings, consistently outperforming existing methods.

1. Introduction

Local clients such as mobile devices or whole organizations generally have limited private data and limited generalizability. Therefore, using all clients' private data to centralized learn a public model will greatly improve performance. However, due to the existence of data silos and data privacy, we cannot use traditional centralized learning in practical applications [19]. To address these challenges, Federated Learning (FL) has been proposed by McMahan *et al.* [33]. Federated learning is a distributed machine learning framework that enables multiple clients to collaboratively train models with decentralized data. The clients never share private data with server ensuring basic privacy. Recently, the widely used federated learning algorithms, e.g., FedAvg [33] and FedProx [29], are based on averaging the model parameters of the participating clients. Most of these federated learning methods [44, 27, 40, 32, 8, 14] are developed based on the assumption that participating client models have the same neural architecture.

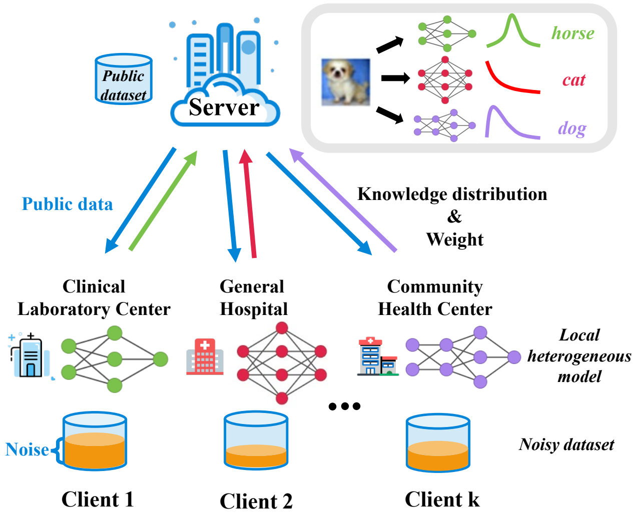


Figure 1. Illustration of federated learning with noisy and heterogeneous clients, where clients possess heterogeneous local models and noisy datasets with different noise rates.

In real-world scenarios, due to the differences in the personalized requirements [28], each client might expect to design its own model independently [17, 38, 41, 26, 13, 3, 9], resulting in the model heterogeneous federated learning problem. For example, when many healthcare organizations engage in collaborative learning without sharing private

[*]Corresponding Author: Mang Ye (yemang@whu.edu.cn)

data, they design different models to meet different tasks and specifications, as illustrated in Fig. 1. In this scenario, institutions are often reluctant to disclose or share the details of the model design due to business purposes. Therefore, to perform federated learning with heterogeneous models, a number of heterogeneous federated learning methods have been proposed [23, 31, 30, 60, 7, 55, 6, 16]. FedMD [23] is a framework based on knowledge distillation, which is implemented through the class scores by client models on the public dataset. FedDF [31] leverage unlabeled data to perform ensemble distillation for each different model architecture. These strategies mainly rely on a unified global consensus or shared models. However, learning a global consensus has a major limitation in that the clients cannot individually adjust their learning direction to accommodate the differences among clients. Besides, building extra models will increase computational overhead, thereby affecting efficiency and effectiveness. Therefore, *how to perform federated learning with heterogeneous clients without relying on a global consensus or shared models* is challenging.

In addition, the methods mentioned above mainly rely on the assumption that each client has a clean dataset, which cannot be satisfied in many practical applications. When the clients contain inevitable noisy samples, existing federated learning methods cannot eliminate the negative effect caused by label noise, suffering from a significant performance drop [25]. Since federated learning contains a large number of participating clients, the data in each participating client usually has different noisy patterns. Generally, in practical applications, the label noise is caused by the following two aspects: 1) Due to the limitation and scarcity of human expertise, the quality of labeled data will be affected by human subjective factors, which means that high-quality labeled data requires high cost and thus inevitably results in some wrong annotation. 2) In the federated learning framework, considering the user fairness issue, there may be some free-riding participants in the system who want to learn from the global model, but do not want to provide useful information. Therefore, some users are reluctant to share their real information with other users and deliberately generate some wrong labels. In order to reduce the negative impact of label noise, existing methods [12, 57, 46, 35, 42, 24, 58, 49, 53, 52] are usually developed for the image classification task with a single model. The approaches can be divided into four categories: label transition matrix estimation [39, 36, 50, 10], robust regularization [54, 2, 34], robust loss function design [43, 5, 48], and clean sample selection [11, 47, 18]. Under the federated learning framework, we expect that each class of samples will be learned sufficiently while avoiding overfitting to noisy samples. Therefore, *how to reduce the negative impact of the internal label noise on the local model convergence during the local update phase* is an important issue.

Furthermore, the above mentioned two problems lead to a new challenging issue, *i.e.*, how to reduce the negative and noisy influence from other clients while collaborative learning in the federated learning framework. Due to model heterogeneity, the participating clients will have different decision boundaries and varying noisy patterns. As a result, besides local noise, we also need to pay attention to the noise from other clients, and then it is crucial to reduce the contribution of noisy clients in the whole federated system. The existing methods for solving noise in machine learning only eliminate the negative effect of internal model label noise, but are unable to solve the noise from other clients. Therefore, *it is crucial to handle the noisy feedback from other noisy clients under the federated learning framework.*

In this paper, we provide the corresponding solution RHFL (Robust Heterogeneous Federated Learning) for the robust federated learning problem with noisy and heterogeneous clients: **1) Aligning the logits output distributions in heterogeneous federated learning.** In order to communicate learning in the presence of model heterogeneity, we learn the knowledge distribution of other clients by aligning models feedback on public data. This allows each client to adjust different learning directions, which does not depend on a public model for communication. **2) Local noise learning with a noise-tolerant loss function.** We analyze the negative effect caused by internal model label noise. In the local learning phase, we consider symmetrically using cross-entropy loss and reverse cross-entropy loss to avoid overfitting noise samples while fully learning all classes. **3) Client confidence re-weighting for external noise.** Since the label noise comes from the feedback by other clients, we propose a new weighting approach, namely Client Confidence Re-weighting (CCR), to reduce the contribution of noisy clients in federated communication. CCR models the loss decreasing pattern of the local models on private datasets for participant re-weighting. Its basic idea is to simultaneously quantifies the label quality of dataset through the loss value and the loss decreasing speed, and then adaptively assign the weight of the clean and efficient clients. The main contributions in this work are as follows:

- We study a novel and important robust federated learning problem with noisy and heterogeneous clients.
- We propose a new loss correction approach CCR, which computes the optimal weighted combination of participating clients. CCR adaptively adjusts the contribution of each client during loss updating, reducing the contribution of noisy clients, and increasing the contribution of clean clients.
- We validate the proposed method under various settings, including both heterogeneous and homogeneous models with different noise types and noise rates. Experimental results show that RHFL consistently achieves stronger robustness than competing methods.

2. Related Work

Federated Learning. The concept of federated learning was first proposed in 2017 by McMahan *et al.* [33]. It is a machine learning setting that allows clients to collaboratively train models while protecting data privacy. McMahan *et al.* propose FedAvg, in which the client uses private data to reduce the local gradient of the local model, and the server uses the averaged model parameters to aggregate the local model. Li *et al.* [29] build a framework similar to FedAvg, but it can adaptively set the local calculations according to different devices and iterations. Wang *et al.* [44] propose to collect the weight of each layer of the client and performs one-layer matching to obtain the weight of each layer of the federated model.

For learning with model heterogeneous clients, Li *et al.* [23] implement communication between models through knowledge distillation. The server collects the class scores of the public data set on each client model and calculates the average value as the updated consensus. Lin *et al.* [31] leverage ensemble distillation for model fusion, and it can be carried out through unlabeled data. Diao *et al.* [4] propose to adaptively allocate a subset of global model parameters as local model parameters according to the corresponding capabilities of the local client. Liang *et al.* [30] introduce an algorithm to jointly train the compact local representation and global model of the client.

In summary, existing methods are usually developed under the assumption that all clients possess clean data without noise, dedicated to making federated learning more efficient, and preserving the privacy of user data. There is no research on mitigating the impact of noise in heterogeneous federated learning.

Label Noise Learning. In machine learning, many methods have been proposed to handle label noise. They can be divided into four main categories: 1) *Label transition matrix* [39, 36, 50]. The main idea is to estimate the probability of each label class flipping to another class. Sukhbaatar *et al.* [39] add a noise layer to the network to make the network output match the noisy label distribution. Patrini *et al.* [36] design an end-to-end loss correction framework that makes recent noise estimation techniques applicable to the multi-class setting. Yao *et al.* [50] transform the noise into a Dirichlet-distributed space, use the dynamic label regression method iteratively infer the potential real labels, and jointly train the classifier and noise modeling. 2) *Robust regularization* [54, 2, 34, 22]. Robust regularization can effectively prevent the model from overfitting to noisy labels. Zhang *et al.* [54] propose Mixup, which trains the convex combination of pairs of samples and their labels to regularize the hybrid neural network. Arpit *et al.* [2] demonstrate regularization can reduce the memory speed of noise without affecting the learning of real data. Miyato *et al.* [34] introduce a regularization method based on virtual adversarial loss, and defined the adversarial direction without label information, which makes it suitable for label noise setting. 3) *Robust loss function* [43, 5, 59, 57]. Some methods achieve robust learning by using noise-tolerant loss functions. Rooyen *et al.* [43] propose a convex classification calibration loss, which is robust on symmetric label noise. Ghosh *et al.* [5] analyze some loss functions that are widely used in deep learning and proved that MAE is robust to noise. 4) Selecting possibly clean samples [11, 47, 18, 51]. The methods select clean samples from the noisy training dataset for learning, or re-weighting for each sample. The core idea is to reduce the attention to noisy-labeled samples in each iteration for training. Han *et al.* [11] propose Co-teaching, which trains two deep neural networks at the same time and selects data with potentially clean labels for cross-trains. Wei *et al.* [47] present JoCoR, which calculates the joint loss with Co-Regularization, and then select small loss samples to update network parameters. Jiang *et al.* [18] introduce MentorNet, which provides a sample weighting scheme for StudentNet, and MentorNet learns a data-driven curriculum dynamically with StudentNet.

Previous methods for solving label noise are mainly under the centralized setting. However, in the federated setting, the server cannot directly access the private datasets of clients. In the model heterogeneous setting, different model architectures will lead to inconsistent decision boundaries and different noisy patterns.

3. Robust Heterogeneous Federated Learning

Problem Definition and Notation. Under the heterogeneous federated learning setting, we consider K clients and one server. We define $\mathbb{C}$ as the collection of all clients, and $|\mathbb{C}| = K$. Therefore, the k-th client $c_k \in \mathbb{C}$ has a private dataset $D_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ with $|x^k| = N_k$. $y_i^k \in \{0,1\}^{N_k}$ indicate the one-hot vector of the ground truth label. In addition, client c_k hold a local model θ_k with different neural architecture, $f(\cdot)$ represents a network, and $f(x^k, \theta_k)$ denotes the logits output of x^k calculated on θ_k. The server cannot access the clients' datasets, and it has a public dataset $D_0 = \{x_i^0\}_{i=1}^{N_0}$,which may belong to different classification tasks from the client datasets. In heterogeneous federated learning, the learning process is divided into a collaborative learning phase and a local learning phase. Moreover, collaborative learning includes T_c epochs and local learning includes T_l epochs. We aim to perform robust federated learning with noisy clients, so suppose that each client has a private noisy dataset $\tilde{D_k} = \{(x_i^k, \tilde{y}_i^k)\}_{i=1}^{N_k}$, where $\tilde{y}_i^k$ represents the noisy label. Because of model heterogeneity, each client has inconsistent decision boundaries and different noisy patterns, which can be formulated as $f(x, \theta_{k_1}) \neq f(x, \theta_{k_2})$. Therefore, except the noise on its own private dataset $\tilde{D}_k$, the client c_k must also pay atten-

模型预训练和KD分开

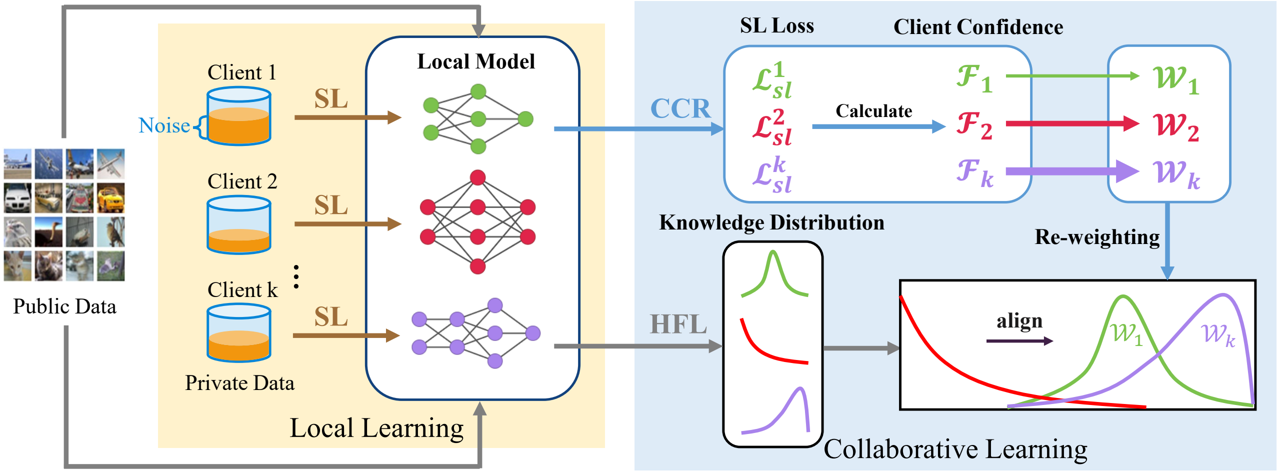


Figure 2. Illustration of RHFL, which performs the heterogeneous FL by aligning the knowledge distributions of individual models on the public dataset §3.1. Under the condition of noisy clients, SL loss is adopted to mitigate overfitting to local noisy data §3.2. As for the noise generated in communication, the weight of noisy clients is lowered by measuring the client confidence §3.3

tion to the noise of other clients $c_{k_0 \neq k}$.

Intuitively, we need to calculate a set of optimal model parameters $\theta_k = \arg\min \mathcal{L}(f(x^k, \theta_k), y^k)$. To achieve this goal, we propose robust federated learning which contains three steps as shown in Fig. 2 : 1) Communicating and learning between heterogeneous clients §3.1. 2) Prevent overfitting to local noisy labels while promoting sufficient learning of all classes §3.2. 3) Individualized weighting of each client to reduce the contribution of noisy clients §3.3.

3.1. Heterogeneous Federated Learning

In collaborative learning phase, we use the public dataset D_0 as a bridge for communication between clients. In the $t_c \in T_c$ epoch of collaborative learning, each client c_k uses the local model $\theta_k^{t_c}$ to calculate the logits output on the public dataset D_0. In this way, the knowledge distribution $R_k^{t_c} = f(D_0, \theta_k^{t_c})$ on the client c_k is obtained. Furthermore, the client uses Kullback–Leibler (KL) divergence to measure the difference in knowledge distribution from other clients.

softmax

KL divergence is usually used to express the difference between two probability distributions. Given two different clients c_{k_1} and c_{k_2}, we fit the knowledge distribution $R_{k_2}^{t_c} = f(D_0, \theta_{k_2}^{t_c})$ with $R_{k_1}^{t_c} = f(D_0, \theta_{k_1}^{t_c})$ in order to measure the difference between the knowledge distribution of c_{k_1} and c_{k_2}, which can be expressed as:

$$\mathcal{KL}(R_{k_1}^{t_c} \parallel R_{k_2}^{t_c}) = \sum R_{k_1}^{t_c} \log\left(\frac{R_{k_1}^{t_c}}{R_{k_2}^{t_c}}\right). \quad (1)$$

The greater the knowledge distribution difference between $R_{k_1}^{t_c}$ and $R_{k_2}^{t_c}$, the more c_{k_1} and c_{k_2} can learn from each other. Therefore, minimizing the KL difference between probability distributions $R_{k_1}^{t_c}$ and $R_{k_2}^{t_c}$ can be considered as a process in which c_{k_1} learns knowledge from c_{k_2}.

In this way, the client c_k calculates the knowledge distribution difference:

$$\mathcal{L}_{kl}^{k,t_c} = \sum_{k_0=1, k_0 \neq k}^{K} KL(R_{k_0}^{t_c} \parallel R_k^{t_c}), \quad (2)$$

where k_0 denotes the clients other than c_k. In Heterogeneous Federated Learning (HFL) method, by measuring the knowledge distribution difference of c_k, all other clients can obtain knowledge from c_k without leakage of data privacy or model design details.

Due to the significant knowledge distribution difference, clients are motivated to perform collaborative learning. Then the clients learn from others by aligning the knowledge distribution:

$$\theta_k^{t_c} \leftarrow \theta_k^{t_c-1} - \alpha \nabla_\theta \left(\frac{1}{K-1} \cdot \mathcal{L}_{kl}^{k,t_c-1}\right), \quad (3)$$

where α represents the learning rate.

3.2. Local Noise Learning

To reduce the negative impact of local noise, we learn the approach mentioned in the Symmetric Cross Entropy Learning [45]. In current Machine Learning, Cross Entropy (CE) loss is one of the most common loss functions, which is deformed according to the KL divergence formula. We denote p and q as the label class distribution and the predicted class distribution, respectively. The KL divergence for p and q is denoted as:

$$\mathcal{KL}(p \parallel q) = \sum p(x) \log(p(x)) - \sum p(x) \log(q(x)), \quad (4)$$

where the first term of formula represents the entropy of p, and the second term represents the cross entropy. Therefore, the CE loss of the sample x can be expressed as:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} p(x_i)\log\left(q(x_i)\right). \tag{5}$$

In the presence of label noise, CE loss shows several limitations. Due to the different levels of simplicity in classes, CE loss cannot make all classes be sufficiently learned or correctly classify all categories. In order to fully converge the difficult-to-learn classes, more rounds of learning will be performed. At this time, the easy-to-learn classes will tend to overfitting the noisy labels, and the overall performance of the model will begin to decline.

In general, the model has the ability to correctly classify samples to some extent. Furthermore, due to the presence of label noise, the prediction result of the model is more reliable than the given label to some extent. Thus p might not be the true class distribution, on the contrary, q reflects the true class distribution to a degree. Automatically, a loss function Reverse Cross Entropy (RCE) [45] based on q is considered, which can align to the class distribution predicted by the model. The RCE loss of the sample x can be expressed as follows:

$$\mathcal{L}_{rce} = -\sum_{i=1}^{N} q(x_i)\log\left(p(x_i)\right). \tag{6}$$

By combining the CE loss and the RCE loss, it is possible to fully learn the difficult-to-learn classes while preventing overfitting noisy labels on the easy-to-learn classes. Then the Symmetric Cross Entropy Learning(SL) loss is formulated as:

添加了一个新的loss项，Reverse Cross Entropy

$$\mathcal{L}_{sl} = \lambda\mathcal{L}_{ce} + \mathcal{L}_{rce}, \tag{7}$$

here λ controls the overfitting of CE to noise. The CE loss strengthens the model fitting effect on each class, the RCE loss is robust to the label noise.

To balance local knowledge and knowledge learned from other clients, we set up a local learning stage. The client will update the local model with its own private dataset to prevent the forgetting of local knowledge. During the training iteration process, the label noise causes the model to update in the wrong direction and eventually fail to converge. To avoid such a result, we adopt SL loss to calculate the loss between the pseudo-label predicted by the model and the corresponding given label. Then, the local update can be denoted as:

监督学习+自监督学习

$$\theta_k^{t_l} \leftarrow \theta_k^{t_l-1} - \alpha\nabla_\theta\mathcal{L}_{sl}^{k,t_l-1}(f(x^k,\theta_k^{t_l-1}),\tilde{y}^k), \tag{8}$$

where $t_l \in T_l$ represents the t_l-th communication round. The client takes advantage of SL loss to update the model when strengthening local knowledge, which can avoid overfitting noisy labels and promote adequate learning.

3.3. Client Confidence Re-weighting

We propose the Client Confidence Re-weighting (CCR) approach to reduce the adverse impact of label noise from other clients during the collaborative learning phase. CCR can individualize the weighting of each client during the communication process to reduce the contribution of noisy clients and pay more attention to clients with clean datasets and efficient models.

To estimate the label quality, SL loss is used to calculate the loss between the predictive output of the local model θ_k on the private noisy dataset $\tilde{D}_k$ and the given label $\tilde{y}^k$. In particular, SL considers both the loss based on the given label and the loss based on the predicted pseudo-label. Thus, a small SL loss $\mathcal{L}_{sl}(f(x^k,\theta_k),\tilde{y}^k)$ indicates that the predicted pseudo-label has a similar distribution to the given label, which means that the private dataset $\tilde{D}_k$ of the client c_k has accurate labels. On the contrary, either a large loss CE based on the given labels $\tilde{y}^k$ or a large loss RCE based on the predicted pseudo-labels $f(x^k,\theta_k)$ will inevitably be a large SL loss. If the loss $\mathcal{L}_{sl}^k$ calculated by the local model θ_k on the private dataset $\tilde{D}_k$ is very large, this signifies that the distribution of the predicted pseudo-labels and the given labels are different, i.e., the private dataset $\tilde{D}_k$ of the client c_k might possess many noisy labels. In this way, the label quality of the dataset $\tilde{D}_k$ can be formulated as:

$$\mathcal{Q}_{t_c}(\tilde{D}_k) = \frac{1}{\frac{1}{N_k}\sum_{i=1}^{N_k}\mathcal{L}_{sl}^{k,t_c}(f(x_i^k,\theta_k),\tilde{y_i}^k)}. \tag{9}$$

To quantify the learning efficiency, we calculate the SL drop rate in each iteration round. The SL drop rate of the client c_k in the T_c iteration is formulated as $\Delta\mathcal{L}_{sl}^{k,t_c}$. Specifically, the loss drop rate can reflect the learning efficiency of the model to some extent. Then we simply quantify the learning efficiency of the client c_k with the SL drop rate:

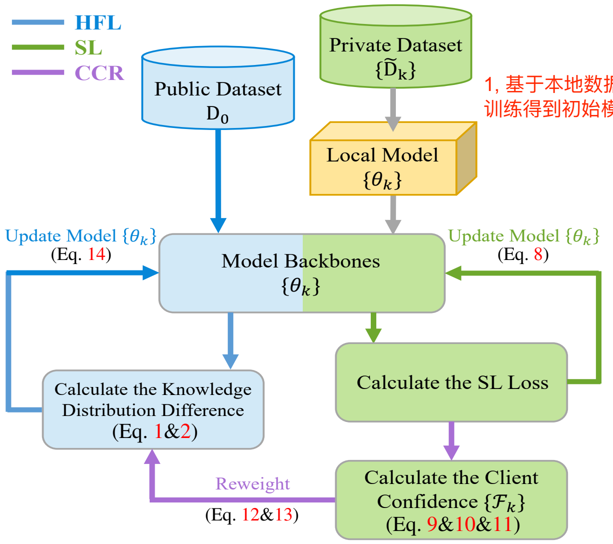
$$\mathcal{P}(\theta_k^{t_c}) = \Delta\mathcal{L}_{sl}^{k,t_c} = L_{sl}^{k,t_c-1} - L_{sl}^{k,t_c}, \tag{10}$$

where $t_c \in T_c$ represents the t_c-th communicate round. By considering both label quality and learning efficiency, the k-th client confidence in round t_c will be defined as:

$$\mathcal{F}_k^{t_c} = \mathcal{Q}_{t_c}(\tilde{D}_k)\cdot\mathcal{P}(\theta_k^{t_c}). \tag{11}$$

It measures the confidence for each client respectively by quantifying the label quality of the private dataset and the learning efficiency of the local model. In the collaborative learning phase, we re-weight each client with its confidence, so that clients can learn more knowledge from confident clients and reduce the learning weight on unconfident clients. The client confidence $\mathcal{F}_k^{t_c}$ determines the weight assigned to client c_k in round t_c, which can denoted as:

$$w_k^{t_c} = \frac{1}{K-1} + \eta\frac{\mathcal{F}_k^{t_c}}{\sum_{k=1}^{K}\mathcal{F}_k^{t_c}}, \tag{12}$$



1, 基于本地数据集训练得到初始模型

Figure 3. The flow chart of RHFL.

$$\mathcal{W}_k^{t_c} = \frac{\exp(w_k^{t_c})}{\sum_{k=1}^{K} \exp(w_k^{t_c})}, \tag{13}$$

here η controls the impact of client confidence $\mathcal{F}$. When $\eta = 0$, the client confidence is not considered. The above weighted regularization can minimize the knowledge of the noisy client with poor label quality and low learning efficiency from being learned, thereby solving the problem of noise from other clients. We dynamically weight the knowledge distribution learned by the client in each round, as:

其他client的Logits

$$\theta_k^{t_c} \leftarrow \theta_k^{t_c-1} - \alpha \nabla_\theta (\mathcal{W}_k^{t_c} \cdot \mathcal{L}_{kl}^{k,t_c-1}). \tag{14}$$

With the iteration of training, each model will be updated in the direction of the clean and efficient clients.

Summary. The entire procedure is summarized in Fig. 3. First, each client c_k updates the local model θ_k with the private noisy dataset $\tilde{D}_k$ to get a set of pre-trained models. In the collaborative learning, the client c_k aligns the feedback distribution of other clients $c_{k_0 \neq k}$, to learn the knowledge from others in Eq. 1&2. In this way, the clients can individually adjust their learning direction according to the differences of models, instead of simply learning a global consensus. Therefore, in order to reduce the impact of local noise, we use SL loss to update the local model in Eq. 8. The loss and the loss drop rate are adopted to measure the label quality of private dataset and the learning efficiency of local model, then calculate the client confidence based on the label quality and the model learning efficiency in Eq. 9&10&11. When learning the knowledge distribution from other clients, the participants are re-weighted according to the client confidence in Eq. 12&13. Through personalized weighting, we adjust the involvement of the noisy client in the federated system and avoid the impact of noise during communication.

Table 1. Prove the effectiveness of heterogeneous federated learning in a no-noise scenario §4.2. θ_k represents the local model of client c_k, Avg denotes the average test accuracy on four models.

Method	θ_1	θ_2	θ_3	θ_4	Avg
w/o FL	82.03	81.85	68.27	77.96	77.53
FedMD [23]	81.36	80.43	72.31	79.89	78.50
FedDF [31]	81.95	81.14	72.17	**80.61**	78.97
RHFL	**82.96**	**82.72**	**73.21**	79.04	**79.48**

4. Experimental

4.1. Experimental Setting

Datasets and Models. Our experiments are conducted on three datasets, Cifar10 [21] and Cifar100 [21], which are widely used in the research of label noise. Here we set public dataset on the server as a subset of Cifar100, and randomly divide Cifar10 to different clients as private datasets.

In the heterogeneous model scenario, we assign four different networks, ResNet10 [15], ResNet12 [15], ShuffleNet [56] and Mobilenetv2 [37], to four client respectively. While in the homogeneous model scenario, the networks of all four clients are set as ResNet12.

Noise Type. We use the label transition matrix $\mathcal{M}$ to add label noise to the dataset, where $\mathcal{M}_{mn} = flip(\tilde{y} = n | y = m)$ represents that the label y is flipped from the clean m class to the noisy n class. There are two widely used structures for the matrix $\mathcal{M}$, symmetric flip [43] and pair flip [11]. Symmetric flip means that the original class label will be flipped to any wrong class labels with equal probability. As for pair flip, it means that the original class label will only be flipped to a very similar wrong category.

Implementation Details. The size of private dataset and public dataset is specified as $N_k = 10,000$ and $N_0 = 5,000$ respectively. We perform $T_c = 40$ collaborative learning epochs for different models. Considering the scale difference between public dataset and private dataset, we set the number of local learning epochs to $T_l = \frac{N_k}{N_0}$, which can better balance the learning of local knowledge and the knowledge from other clients. Furthermore, we use the Adam [20] optimizer with an initial learning rate of $\alpha = 0.001$ and the batch size of 256. λ set as 0.1 and η set as 0.5. Due to this paper mainly focuses on the robustness of the federated learning on noisy supervision, we set the noise rate $\mu = 0.1$ and 0.2, discussing the results under both symmetric flip and pair flip noise types. To generate the noisy dataset $\tilde{D}$, we flip 20% of the labels in the training dataset of Cifar10 [21] to the wrong labels and keep the test dataset of Cifar10 constant to observe the model robustness. The c_k client randomly selects N_k samples from the shuffled Cifar10, so the client may have different proportions of noise, which is more in line with the actual situation. All experiments are supported by Huawei MindSpore [1].

State-of-the-Art Methods. In order to prove the effectiveness of RHFL in the heterogeneous model scenario,

Table 2. Ablation study §4.2 with the noise rate $\mu = 0.1$, θ_k represents the local model of the client c_k.

Components			Pairflip					Symflip				
HFL	SL	CCR	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
			77.98	76.75	66.89	74.33	73.99	76.20	76.05	64.96	74.31	72.88
✓			73.64	76.02	68.22	**76.64**	73.63	74.05	76.48	66.91	73.74	72.80
	✓		76.23	76.96	63.35	68.79	71.33	77.20	78.06	64.27	70.71	72.56
✓	✓		78.41	**79.38**	68.05	74.86	75.18	**78.81**	76.68	67.42	75.64	74.64
✓	✓	✓	**78.86**	78.76	**69.60**	71.83	**74.76**	78.40	**78.36**	**69.47**	**76.93**	**75.79**

Table 3. Ablation study §4.2 with the noise rate $\mu = 0.2$, θ_k represents the local model of the client c_k.

Components			Pairflip					Symflip				
HFL	SL	CCR	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
			72.31	71.84	61.78	69.67	68.90	72.01	70.15	59.62	69.42	67.80
✓			68.43	66.57	60.67	70.36	66.51	68.08	64.43	62.09	70.46	66.26
	✓		73.94	73.18	61.78	68.18	69.27	75.69	73.81	60.29	68.64	69.61
✓	✓		74.62	74.20	62.45	72.55	70.96	78.05	**77.80**	63.41	**74.14**	**73.35**
✓	✓	✓	**77.81**	**76.09**	**66.61**	**72.78**	**73.32**	**78.14**	76.77	**64.23**	73.90	73.26

we compare RHFL with the heterogeneous FL algorithm FedMD [23] and FedDF [31] under the same settings. FedMD is based on knowledge distillation, in which each client computes the class scores on public data and then approaches the consensus. FedDF builds a distillation framework for robust federated model fusion, which allows for heterogeneous models and data. To demonstrate the validity of RHFL in the homogeneous model case, we compare it with FedAvg [33], FedMD, and FedDF. FedAvg leverages the private dataset for local gradient descent, followed by the server aggregating the updated model on average. Since our setting is not the same as theirs, we use the key of these algorithms for our experiments.

4.2. Ablation Study

Effect of Each Component. We first evaluate the effect of each component on two noise rates (0.1, 0.2) with two noise types (pairflip, symflip) in the heterogeneous model scenario, to prove the effectiveness of each component.

1) Effectiveness of HFL: According to Table 1, we observe that in the no-noise scenario, the average accuracy of the four models without communication reaches 77.53%. After adding HFL, the clients conduct federated communication and learn more global knowledge. It is obvious from the results that the testing accuracy of each local model has been effectively enhanced, specifically the average accuracy has increased to 79.48%. In the noisy scenario, the effect of adding HFL will have some degree of degradation than without FL in Table 2&3. Then, we analyze the reasons for this phenomenon. Since the existence of noise leads to degradation of model performance, HFL causes the clients to keep communicating learning the wrong knowledge and updating the model in the wrong direction. Therefore, it is essential to focus on noisy data in heterogeneous FL.

2) Effectiveness of SL: We add the SL loss component to the baseline to avoid the influence of noisy data during the local update phase. When the noise rate is 0.1, the average test accuracy does not improve obviously in Table 2. However, when the noise rate is 0.2, the performance of most models has been significantly improved in Table 3. From the above phenomenon, it can be inferred that the higher the noise rate, the better the performance of SL loss. Our analysis is due to the fact that the reliance on the given label is reduced in SL loss, but the truthfulness of the given label is high in the setting of low noise rate, which diminishes the effectiveness of SL loss to some extent. This proves that at high noise rates, the addition of SL loss enables the models to correctly learn the local true distribution. We add the SL loss component on the basis of adding HFL, each model has been remarkably optimized, especially when the noise type is symflip. When the noise rate is 0.2, the improvement is most obvious, the average test accuracy is improved from 66.26% to 73.35%. The SL loss component effectively revises the learning direction of HFL.

3) Effectiveness of CCR: We add the CCR component so that the robustness against noisy data from other clients is enhanced. As shown in Table 3, each model has achieved better performance. The most obvious is that when the noise type is pairflip and the noise rate is 0.2, the average test accuracy of models has increased from 70.96% to 73.32%. CCR enables each client to learn each local knowledge individually, rather than learning global knowledge at the average level, avoiding knowledge learning on noisy clients. This verifies the rationale of re-weighting the models by modeling the loss decreasing curve.

Comparison of Different Noise Rates and Noise Types. As shown in the Table 2&3, we have compared the effectiveness of RHFL under different noise rates and types.

1) Noise Rate: Our method can achieve great overall accuracy under different noise rates, and the improvement of our method will be more obvious at high noise rates. When the noise rate $\mu = 0.1$, our method improves on the original baseline by 0.77% under pairflip noise, and 2.91% under symflip noise. When the noise rate $\mu = 0.2$, our method

Table 4. Compare with the state-of-the-art method §4.3 when the noise rate $\mu = 0.1$, θ_k represents the local model of the client c_k.

Method	Pairflip					Symflip				
	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	77.98	76.75	66.89	74.33	73.99	76.20	76.05	64.96	74.31	72.88
FedMD [23]	74.98	76.89	67.10	**76.64**	73.90	73.23	73.66	67.72	75.54	72.54
FedDF [31]	76.26	75.51	68.41	76.04	74.06	72.07	75.18	67.38	74.47	72.28
RHFL	**78.86**	**78.76**	**69.60**	71.83	**74.76**	**78.40**	**78.36**	**69.47**	**76.93**	**75.79**

Table 5. Compare with the state-of-the-art method §4.3 when the noise rate $\mu = 0.2$, θ_k represents the local model of the client c_k.

Method	Pairflip					Symflip				
	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	72.31	71.84	61.78	69.67	68.90	72.01	70.15	59.62	69.42	67.80
FedMD [23]	68.00	67.81	65.67	**74.02**	68.88	67.31	68.54	**64.48**	71.75	68.02
FedDF [31]	68.66	69.68	62.36	72.12	68.21	67.36	68.56	63.60	70.83	67.59
RHFL	**77.81**	**76.09**	**66.61**	72.78	**73.32**	**78.14**	**76.77**	64.23	**73.90**	**73.26**

Table 6. Compare with the state-of-the-art method §4.3 in the setting of the homogeneous model, we set all four local models as ResNet12, and took the average test accuracy of the local models for demonstration. μ denotes the noise rate.

Method	$\mu = 0.1$		$\mu = 0.2$	
	Pairflip	Symflip	Pairflip	Symflip
Baseline	75.16	71.61	66.14	66.87
FedAvg [33]	79.73	77.32	73.57	74.90
FedMD [23]	74.96	75.57	69.32	70.52
FedDF [31]	77.76	75.90	69.98	71.03
RHFL	**81.03**	**79.60**	**77.85**	**78.83**

improves on the original baseline by 4.42% under pairflip noise, and 5.46% under symflip noise. So when there are more noisy labels in the dataset, using our method can achieve a more significant improvement. This is likely because in the high-noise-rate scenario, the label distribution is more different from the true distribution, and the addition of the SL component makes each client more convergent to learn the pseudo-label distribution in the process of local update. The addition of the CCR component makes the knowledge of relatively noisy clients be learned less in the process of collaborative learning, so our approach demonstrates more significant effectiveness.

2) Noise Type: Our method shows good robustness regardless of the noise generated by symflip or pairflip transition matrix. According to Table 2&3, we observe that our method performs slightly better under symflip noise.

4.3. Comparison with State-of-the-Art Methods

Heterogeneous Federated Learning Methods. We compare with the state-of-the-art heterogeneous FL method under the same setting. The baseline refers to the method in which the client trains local model on private dataset without FL. Therefore, the comparisons on two noise rates (0.1, 0.2) are shown in Table 4&5. The experiments demonstrate that our proposed method outperforms the existing strategies under various noise settings. As the noise rate rises from 0.1 to 0.2, it can be seen that the average test accuracy of FedMD [23] and FedDF [31] drops significantly, by 5.02% for FedMD and 5.85% for FedDF on pairflip noise, and by 4.52% for FedMD and 4.69% for FedDF on symflip noise. As for RHFL, it drops 1.44% on pairflip noise and 2.53% on symflip noise. The above can prove that our proposed solution is robust against different noise settings.

Homogeneous Federated Learning Methods. We compare RHFL with the state-of-the-art FL methods under model homogeneous setting in Table 6, demonstrating its superiority over existing federated algorithms in handling noise. The results show that our method has achieved remarkable results in the model homogeneous scenario, especially when the noise type is symflip and the noise rate is 0.2, our method improves by 11.96% on the basis of the baseline. In this case, the average test accuracy of our method reaches 78.83%, while the best-performing existing method, FedAvg [33], achieves only 74.90%.

5. Conclusion

This paper studies a novel problem of how to perform robust federated learning with noisy heterogeneous clients. To address this issue, a novel solution RHFL is proposed. We align the feedback distribution on public data to enable federated learning between model heterogeneous clients. To avoid each model overfitting to noise during the local learning, a symmetric loss is adopted to update the local model. For the noisy feedback from other participants, we propose a flexible re-weighting method CCR, which effectively avoids overlearning from noisy clients and achieves robust federated collaboration. Extensive experiments prove the effectiveness of each component included in our approach. Moreover, we demonstrate that our method achieves higher accuracy than the current state-of-the-arts under both model homogeneous and heterogeneous scenarios.

Acknowledgement. This work is supported by National Natural Science Foundation of China (62176188), Key Research and Development Program of Hubei Province (2021BAA187), Zhejiang Lab (NO.2022NF0AB01), and CAAI-Huawei MindSpore Open Fund.

References

[1] Mindspore, https://www.mindspore.cn/, 2020. 6

[2] Devansh Arpit, Stanisław Jastrzebski, 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 *ICML*, pages 233–242, 2017. 2, 3

[3] Ayush Chopra, Surya Kant Sahu, Abhishek Singh, Abhinav Java, Praneeth Vepakomma, Vivek Sharma, and Ramesh Raskar. Adasplit: Adaptive trade-offs for resource-constrained distributed deep learning. *arXiv preprint arXiv:2112.01637*, 2021. 1

[4] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *ICLR*, 2020. 3

[5] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, pages 1919–1925, 2017. 2, 3

[6] Gautham Krishna Gudur, Bala Shyamala Balaji, and Satheesh K Perepu. Resource-constrained federated learning with heterogeneous labels and models. In *ACM SIGKDD Workshop*, 2020. 2

[7] Gautham Krishna Gudur and Satheesh K Perepu. Federated learning with heterogeneous labels and models for mobile activity monitoring. In *NeurIPS Workshop*, 2020. 2

[8] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *CVPR*, pages 2423–2432, 2021. 1

[9] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *JNCA*, 2018. 1

[10] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, pages 5841–5851, 2018. 2

[11] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 2, 3, 6

[12] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, pages 5138–5147, 2019. 2

[13] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. In *NeurIPS*, page 14068–14080, 2020. 1

[14] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Towards non-iid and invisible data with fednas: Federated deep learning via neural architecture search. In *CVPR Workshop*, 2020. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[16] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, 2022. 2

[17] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *ICLR*, 2021. 1

[18] 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 *ICML*, pages 2304–2313, 2018. 2, 3

[19] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 1

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 6

[22] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 3

[23] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. In *NeurIPS Workshop*, 2019. 2, 3, 6, 7, 8

[24] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, pages 5051–5059, 2019. 2

[25] Li Li, Huazhu Fu, Bo Han, Cheng-Zhong Xu, and Ling Shao. Federated noisy client learning. *arXiv preprint arXiv:2106.13239*, 2021. 2

[26] Qinbin Li, Bingsheng He, and Dawn Song. Model-agnostic round-optimal federated learning via knowledge transfer. *arXiv preprint arXiv:2010.01017*, 2020. 1

[27] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021. 1

[28] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, pages 50–60, 2020. 1

[29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 1, 3

[30] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. In *NeurIPS Workshop*, 2020. 2, 3

[31] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *NeurIPS*, page 2351–2363, 2020. 2, 3, 6, 7, 8

[32] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *CVPR*, pages 1013–1023, 2021. 1

[33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-

efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017. 1, 3, 7, 8

[34] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, pages 1979–1993, 2018. 2, 3

[35] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017. 2

[36] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017. 2, 3

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 6

[38] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020. 1

[39] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *ICLR*, 2015. 2, 3

[40] Jingwei Sun, Ang Li, Binghui Wang, Huanrui Yang, Hai Li, and Yiran Chen. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *CVPR*, pages 9311–9319, 2021. 1

[41] Lichao Sun and Lingjuan Lyu. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537*, 2020. 1

[42] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018. 2

[43] Brendan Van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*, pages 10–18, 2015. 2, 3, 6

[44] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020. 1, 3

[45] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, pages 322–330, 2019. 4, 5

[46] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020. 2

[47] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020. 2, 3

[48] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, pages 6222–6233, 2019. 2

[49] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. *arXiv preprint arXiv:2012.01700*, 2020. 2

[50] Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W Tsang, and Jun Sun. Safeguarded dynamic label regression for noisy supervision. In *AAAI*, pages 9103–9110, 2019. 2, 3

[51] Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *ICML*, pages 10789–10798, 2020. 3

[52] Mang Ye, He Li, Bo Du, Jianbing Shen, Ling Shao, and Steven CH Hoi. Collaborative refining for person re-identification with label noise. *IEEE TIP*, 2021. 2

[53] Mang Ye and Pong C Yuen. Purifynet: A robust person re-identification model with noisy labels. *IEEE TIFS*, 2020. 2

[54] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3

[55] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *NeurIPS*, 2021. 2

[56] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 6

[57] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8792–8802, 2018. 2, 3

[58] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H. Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, pages 1237–1246, 2019. 2

[59] Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *ICML*, 2021. 3

[60] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, 2021. 2