



Federated Learning with Label-Masking Distillation

Jianghu Lu

Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, UCAS
lujianghu@iie.ac.cn

Shikun Li

Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, UCAS
lishikun@iie.ac.cn

Kexin Bao

Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, UCAS
baokexin@iie.ac.cn

Pengju Wang

Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, UCAS
wangpengju@iie.ac.cn

Zhenxing Qian

School of Computer Science, Fudan
University
zxqian@fudan.edu.cn

Shiming Ge*

Institute of Information Engineering,
Chinese Academy of Sciences
School of Cyber Security, UCAS
geshiming@iie.ac.cn

ABSTRACT

Federated learning provides a privacy-preserving manner to collaboratively train models on data distributed over multiple local clients via the coordination of a global server. In this paper, we focus on label distribution skew in federated learning, where due to the different user behavior of the client, label distributions between different clients are significantly different. When faced with such cases, most existing methods will lead to a suboptimal optimization due to the inadequate utilization of label distribution information in clients. Inspired by this, we propose a label-masking distillation approach termed *FedLMD* to facilitate federated learning via perceiving the various label distributions of each client. We classify the labels into majority and minority labels based on the number of examples per class during training. The client model learns the knowledge of majority labels from local data. The process of distillation masks out the predictions of majority labels from the global model, so that it can focus more on preserving the minority label knowledge of the client. A series of experiments show that the proposed approach can achieve state-of-the-art performance in various cases. Moreover, considering the limited resources of the clients, we propose a variant *FedLMD-Tf* that does not require an additional teacher, which outperforms previous lightweight approaches without increasing computational costs. Our code is available at <https://github.com/wnma3mz/FedLMD>.

CCS CONCEPTS

• Computing methodologies → Distributed algorithms.

KEYWORDS

Federated Learning, Knowledge Distillation

*Shiming Ge is the corresponding author (geshiming@iie.ac.cn).



This work is licensed under a Creative Commons Attribution International 4.0 License.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0108-5/23/10.
<https://doi.org/10.1145/3581783.3611984>

ACM Reference Format:

Jianghu Lu, Shikun Li, Kexin Bao, Pengju Wang, Zhenxing Qian, and Shiming Ge. 2023. Federated Learning with Label-Masking Distillation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*, October 29–November 3, 2023, Ottawa, ON, Canada. ACM, Lisbon, Portugal, 11 pages. <https://doi.org/10.1145/3581783.3611984>

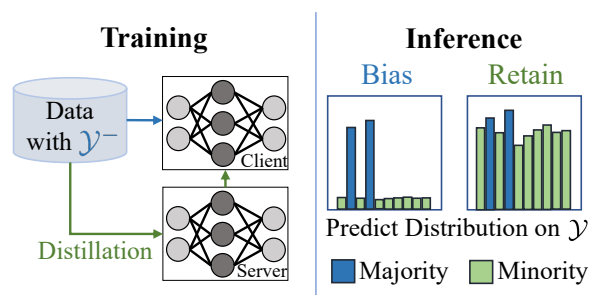


Figure 1: The model trained on the private dataset of a client with partial class labels \mathcal{Y}^- is generally biased to \mathcal{Y}^- due to knowledge missing over complete class labels \mathcal{Y} . Our FedLMD method proposes to alleviate it by utilizing the global model from the server to retain the knowledge of minority labels $\mathcal{Y} \setminus \mathcal{Y}^-$.

1 INTRODUCTION

The development of multimedia technology and its various emerging commercial applications have sparked global discussions on the ethics of artificial intelligence [21]. Among these discussions, privacy and security issues have become a key concern for society [29]. Artificial intelligence technology relies heavily on user data uploaded to central servers, which could lead to the leakage of sensitive personal data [30]. The centralized collection and use of massive personal data pose serious threats to individual privacy. Once the data are breached or misused, the consequences can be devastating. Additionally, countries worldwide have enacted laws and regulations, such as the European Union's General Data Protection Regulation (GDPR), to restrict such behavior [2, 32]. Therefore, the multimedia field needs to improve the centralized model training method to gain public recognition and address such concerns.

Federated learning (FL) [28] has been proposed to provide a feasible solution to jointly train models on distributed data from multiple parties or clients in a privacy-preserving manner. It generally applies a server as coordinator to communicate parameters (gradients or weights of the model) between each client and server, realizing the knowledge sharing rather than data among clients. Since the data only stays local, it is considered to be a privacy-preserving algorithm. It has shown promising results in multimedia applications such as person re-identification [51, 52], medical images [24, 27], emotion prediction [33] and deepfake detection [12].

In classical FL algorithm FedAvg, the uploaded model parameters are weighted and averaged to implicitly exchange the knowledge of each client. It can work well when the data distributions are identical in clients. However, the realistic data distribution usually is different across clients [14], *i.e.*, non-independent isodistribution (Non-IID). It means that the optimization goals of various client models are much different, and the server-side model is much more difficult to optimize, and may even fail to converge [23]. In this paper, we focus on a more specific case, *i.e.*, label distribution skew. For instance, diseases can be simply divided into several class labels according to severity, and small clinics in rural areas usually have more examples of minor diseases but fewer or no examples of severe diseases compared to large hospitals. For convenience, we call this realistic scenario the label heterogeneity case.

To address the challenge, some researchers improve FedAvg in terms of weight assignment during aggregation and model aggregation way on the server-side [11, 13, 26, 42]. While compared to server-side optimization, client-side optimization is often more effective and straightforward because the data resides on the client-side. The existing client-side methods usually regularize the constraints on the model output or the model parameters themselves [1, 19, 20, 23, 43, 45]. Although these methods can alleviate the challenge in a certain extent, they don't effectively utilize the useful information of varying label distributions in clients under large label heterogeneity, leading to a suboptimal optimization. And this information is crucial and determines the severity of label heterogeneity. Thus, it is necessary to explore an effective solution that can address a key problem: how to exploit the information of label distributions in various clients to perform stable and effective FL?

By revisiting the training process on a particular client in the classical FL Fig. 1, the learned model is prone to be biased toward the majority class labels and forget the absent (or minority) class labels under the label heterogeneity case. In order for the model to learn about minority labels without additional communication, we propose an approach named **Label-Masking Distillation** for federated learning (FedLMD) via perceiving the label distributions of each client. The knowledge distillation (KD) has been shown to extract dark knowledge from models and thus reduce the risk of catastrophic forget in FL [8, 9, 19, 46]. As in the previous study, we use the local model as the student, while the global model is updated based on multiple client models. Thus, it is considered as the teacher with more comprehensive label knowledge. To achieve a more effective distillation process, we employ label masking distillation on the client-side model. We classify the labels into majority and minority labels based on the number of examples per label during training. The model can easily learn the knowledge of the majority labels, because of they have sufficient samples. However,

the knowledge of the minority labels is prone to being forgotten by the model [19]. Therefore, to preserve minority knowledge in the model, we only distilled the minority part of the global model to the client-side model. Specifically, we decouple the logits of the global model into two parts: majority and minority, and mask out the majority part of the global logits. Overall, the client-side model learns the knowledge from two sources: majority from the local data and minority from the global model.

When FL is deployed in real-world applications, the client-side resources have to be seriously considered [37]. Therefore, we further optimize the computational cost and storage space of the proposed approach. We found that a teacher model with poor performance can still help local models in FL. So we replace the teacher logits with a fixed vector, as demonstrated in [48]. Since it does not require an additional teacher model, we named it FedLMD-Tf.

In summary, our main contributions are three folds.

- We revisit the problem caused by label heterogeneity through a simple experiment and find that the main reason why local models are prone to be biased is the lack of supervision information from minority labels.
- We propose FedLMD under the label heterogeneity case. By decoupling the logits of the teacher model and masking out the majority part, the proposed approach is able to retain the forgotten label knowledge for clients by distilling knowledge from the minority part.
- We conduct a series of sufficient experiments to show that FedLMD outperforms the state-of-the-art methods on classification accuracy and convergence speed. We also propose FedLMD-Tf which consistently outperforms previous lightweight federated learning methods.

2 RELATED WORK

Federated Learning on Non-IID Data. One of FL's current significant challenges, data heterogeneity, can lead to difficulties in model convergence [22]. The optimization can be done from the server-side and the client-side respectively. For server-side optimization, they focus on improving the robustness of the global model by improving the aggregation method [11, 34, 42, 44, 47, 49].

The optimization in client focuses on constraining model update to avoid catastrophic forgetting. FedProx [23] constrains the optimization of local model by computing L2 loss between the local model and global model parameters. Similarly, FedDyn [1] and FedCurv [39] are improved based on the relationship between the model parameters. SCAFFOLD [15] corrects the local updates by introducing control variates and they are also updated by each client during local training. On this basis, FedNova [43] achieved automatically adjusts the aggregated weight and effective local steps according to the local progress. Unlike these methods, FedRS [25] adds the scaling factor to SoftMax function using information about the distribution of the data to restrain the update of the parameters of the constrained model updates to the missing classes.

Knowledge Distillation in Federated Learning. KD [10] is considered to be able to extract dark knowledge from the teacher. It can be optimized from both server-side and client-side perspectives. Some researchers exploit the feature of multiple models on the

server-side of FL to perform integrated multi-teacher KD [3, 4, 6, 26, 31, 36, 38, 40, 41].

From the perspective of client-side, some studies use data-free KD to expand the local dataset to ensure that the model has access to sufficient data examples during training [49, 50]. However, they cause additional communication overhead and may also result in privacy leakage. Alternatively, KD can improve the performance of the local model by extracting the dark knowledge of the global model. MOON [20] constrains the model training by constructing the contrastive loss between the local model and the global model. FedNTD [19] mitigates the catastrophic forgetting of the global model by removing the target label when the global model is used as a teacher-distilled local student model. While they effectively mitigate the challenge of data heterogeneity and do not introduce additional communication overhead as well as privacy risks, they impose additional computational overhead on the client-side.

In particular, it should be noted that FedNTD [19] is the most similar to our approach. FedNTD preserves global knowledge, while our approach focuses more on preserving minority label knowledge corresponding to the forgetting of each client. Unlike FedNTD, which only masks out the target class in the teacher model output, we mask out the locally majority labels in the teacher model output from the perspective of label distribution. This achieves more effective knowledge retention. And considering the problem of limited client-side resources, we update the proposed approach to the lightweight version with no additional overhead.

3 CHALLENGE REVISITING

To better understand the challenge caused by label heterogeneity, we first experimentally revisit the problem encountered by FedAvg during the training process¹. The results are shown in Fig. 2, where the darker the color is, the greater the number of samples for the corresponding label is. Fig. 2 (Top), we present the total number of training examples for each label in the uploaded clients under different communication rounds, which clearly shows that the label distribution varies a lot during training.

Fig. 2 (Middle), we show the prediction distribution of the FedAvg method under different rounds, which reflects the instability of its optimization. It can be noticed that the class labels with the most training examples severely affect the prediction distribution, making the model biased toward the majority of class labels and forgetting the minority class labels. Specifically, in the 9-th round, when class label 7 has the most examples, then the prediction distribution of the model is largely biased toward class label 7. Although the model is relatively less affected by the heterogeneity at the later stage of training (e.g., after 100 rounds), the bias toward majority labels still exists. Therefore, it can be found that the main reason why local models are prone to be biased under such cases is the *lack of supervision information from minority class labels*, which inspires us to introduce the information of minority class labels into supervision.

By perceiving the label distributions, as shown in Fig. 2 (Bottom), our FedLMD approach can well resist the bias of majority labels, leading to stable and effective optimization. It can see that the color

depth of different labels tends to be the same at the later stage of training. It means that the prediction label distribution achieved by our FedLMD approach is close to the uniform distribution.

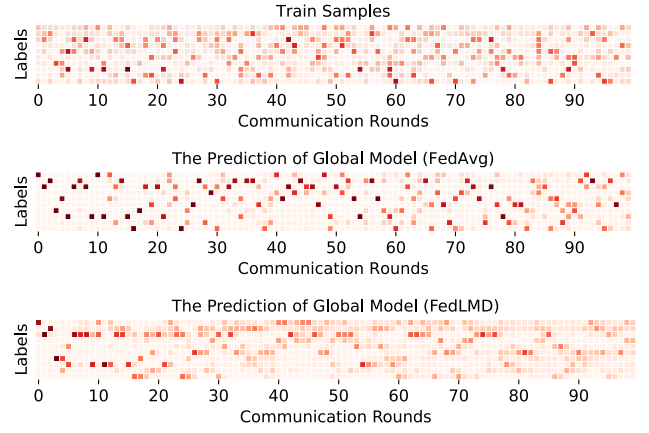


Figure 2: The label distribution of the training examples (Top), the prediction distribution of the FedAvg (Middle), and the prediction distribution of the FedLMD (Bottom) under different communication rounds.

4 PROPOSED METHOD

4.1 Problem Setting

We consider a classical supervised FL system that contains a server and K clients. For the k -th client, it has a local dataset $\mathcal{D}_k = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_k}$, where as $(\mathbf{x}_i, \mathbf{y}_i) \in (\mathcal{X}_k, \mathcal{Y}_k)$ and the weight parameters of model is \mathbf{w}_k . The goal of FL is to obtain a global model by jointly training all clients as follows:

$$\min_{\mathbf{w}_g} \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \mathcal{L}_k(\mathbf{w}_g; \mathcal{D}_k) \quad (1)$$

where \mathbf{w}_g is the weight of the global model and \mathcal{L}_k is the loss function for training the k th client model. On the server-side, the FL system aggregates all uploaded model weights. In each communication round, the clients are specified in \mathcal{K} to train and upload parameters, where $|\mathcal{K}|$ is the number of models to upload.

As mentioned before, label heterogeneity in clients can make the local model biased to the majority labels, leading to unstable and poor optimization. Our goal is to facilitate stable and effective FL via perceiving the various label distributions of each client.

4.2 Label-Masking Distillation

First of all, we divide all class labels into majority labels and minority labels. When $n_{k,y} \geq \frac{n_k}{|\mathcal{Y}_k|}$, class label y is a majority label in the k -th client. The n_k is the total number of samples for all classes and $n_{k,y}$ is the number of samples for class y of the k -th client. In this section, we assume the majority labels are all in \mathcal{Y}_k on k -th client for the sake of convenient expression.

For a training example $(\mathbf{x}, \mathbf{y}) \in (\mathcal{X}_k, \mathcal{Y}_k)$, let the output of the k -th local model as p_k , the output of the global model as p_g , and $\mathbf{1}_y$ is the one-hot vector form of \mathbf{y} . KD [10] is to achieve dark knowledge

¹The specific experimental setup can be found at Sec. 5.1 and we only show the results of the first 100 rounds here.

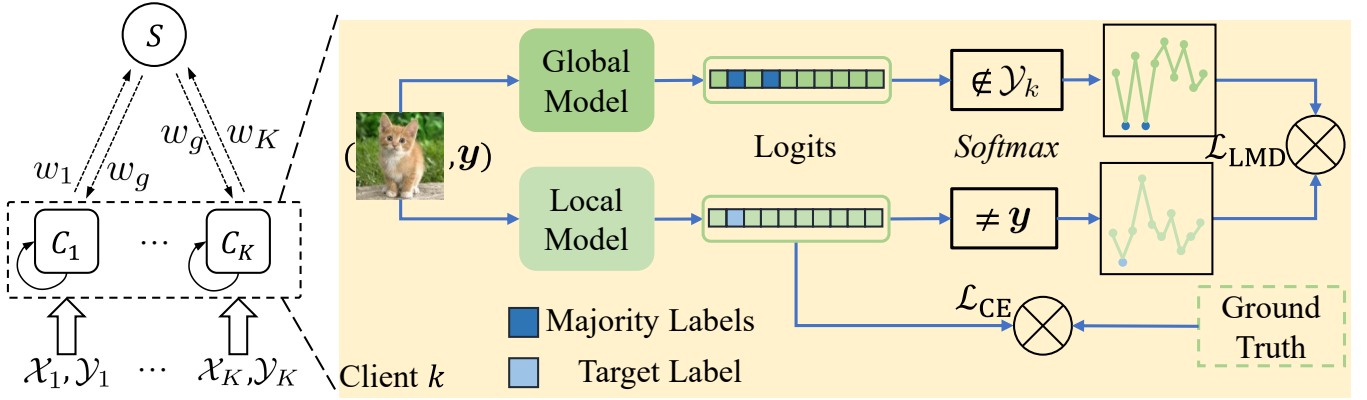


Figure 3: The framework of our approach. For the aggregation process, for the uploaded weight w_1, \dots, w_K of the model are calculated as weighted averages to obtain w_g . For each client, the training loss is the combination of the cross-entropy loss \mathcal{L}_{CE} for learning from local data and the label-masking distillation loss \mathcal{L}_{LMD} for distilling from the global model.

transfer by making the output of the student mimic the output of the teacher. Since the global model is updated based on multiple client models, we use the local model as the student and regard the global model as the teacher with more comprehensive label knowledge, the k -th client's loss can achieve the aim of knowledge retention as follows:

$$\mathcal{L}_k = \mathcal{L}_{CE}(p_k, \mathbf{1}_y) + \beta \mathcal{L}_{KD}(p_k, p_g), \quad (2)$$

where \mathcal{L}_{CE} is the cross-entropy loss for learning the majority labels knowledge, and \mathcal{L}_{KD} is the distillation loss for retaining all the labels knowledge. Here, we fix the weight of \mathcal{L}_{CE} to 1 and β is used as a weighting factor to control the distillation loss.

Although the \mathcal{L}_{KD} can learn from k -th data and assist the bias toward minority labels, it performs the regularization without considering the varying label distributions across clients, leading to a suboptimal optimization. Hence, we improve it by enhancing the KD for minority labels via perceiving the label distributions. We decouple the logits of the global model into two parts: majority and minority. The majority part of teacher logits corresponds to majority labels, and naturally, the minority part corresponds to minority labels. We focus on the minority part of the teacher logits for distillation by masking out the majority part. Because of the majority labels knowledge can be learned from \mathcal{L}_{CE} . This leads to a modified teacher distribution p'_g as:

$$p'_g(i|\mathbf{x}) = \begin{cases} \frac{\exp(z_{g,i}/\tau)}{\sum_{i=1, i \notin \mathcal{Y}_k}^C \exp(z_{g,i}/\tau)} & , \quad i \notin \mathcal{Y}_k \\ 0 & , \quad i \in \mathcal{Y}_k \end{cases}, \quad (3)$$

where $z_{g,i}$ is the logits of the global model for i -th class label, and τ is a temperature factor. We mask out the majority labels for teacher logits (set to 0), which encourages the student model to learn from the knowledge of the minority labels or not all labels, and helps prevent forgetting this knowledge.

For the student model's predictions p_k , a straightforward way is to leave it unchanged. However, this leads to a conflict between \mathcal{L}_{CE} and distillation loss. Because of the teacher's logits for the target label is 0 in distillation (Eq. 3) and the one-hot vector $\mathbf{1}_y$ is

1 in \mathcal{L}_{CE} . Therefore, we mask out the target label in the student model to avoid such conflicts. Additionally, for the majority not-target labels, the student's performance can be further improved by learning from negative supervision [7, 16]. Therefore, we modify the distribution from the student model as:

$$p'_k(i|\mathbf{x}) = \begin{cases} \frac{\exp(z_{k,i}/\tau)}{\sum_{i=1, i \neq y}^C \exp(z_{k,i}/\tau)} & , \quad i \neq y \\ 0 & , \quad i = y \end{cases}, \quad (4)$$

where $z_{k,i}$ is the logits of the k -th client model for i -th class label.

Then, the improved loss can be proposed as follows:

$$\mathcal{L}_k = \mathcal{L}_{CE}(p_k, \mathbf{1}_y) + \beta \mathcal{L}_{LMD}(p'_k, p'_g), \quad (5)$$

where the label-masking distillation loss \mathcal{L}_{LMD} is defined as the Kullback-Leibler divergence between p'_k and p'_g :

$$\mathcal{L}_{LMD}(p'_k, p'_g) = - \sum_{i=1}^C p'_g(i|\mathbf{x}) \log \frac{p'_k(i|\mathbf{x})}{p'_g(i|\mathbf{x})}. \quad (6)$$

And the framework of FedLMD can be seen in Fig. 3.

4.3 Teacher-free Variant

In practical scenarios, the clients may have limited storage space and computation resource. FedLMD introduces an extra model for each client, which will undoubtedly increase the hardware overhead of the client. Therefore, we consider dropping the teacher model to avoid the cost.

The teacher model in distillation generally needs to be pretrained so that they can better provide knowledge to the student model. However, FL is an online learning, *i.e.*, the teacher model does not have good performance in the beginning stage. Inspired by [48], we treat distillation as the label smoothing (LS) regularization by introducing a fixed minority label distribution to replace the output of the teacher model. Specifically, we replace p'_g with μ_k in Eq. 7 as follows:

$$\mathcal{L}_k = \mathcal{L}_{CE}(p_k, \mathbf{1}_y) + \beta \mathcal{L}_{LMD}(p'_k, \mu_k). \quad (7)$$

Algorithm 1 FedLMD and FedLMD-Tf. T is the number of communication rounds, E the local epochs, and η the learning rate. Indices k denote K clients with local dataset \mathcal{D}_k ; w_g^t and w_k^t are the global and k -th client model weights at round t ; \mathcal{K} is the set of selected clients per round.

```

1: Initialization:  $w_g^0$ 
2: for each round  $t = 1, 2, \dots, T$  do
3:   Broadcasts  $w_k^t \leftarrow w_g^{t-1} (k \in [1, \dots, K])$ 
4:    $\mathcal{K} \leftarrow$  a random subset of the  $K$  clients.
5:   for each client  $k \in \mathcal{K}$  in parallel do
6:     for local training steps  $e = 1, \dots, E$  do
7:       // Using Eq. 5 for FedLMD or Eq. 7 for FedLMD-Tf
8:        $w_k^t = w_k^t - \eta \nabla_w \mathcal{L}_k(w_k^t, \mathcal{D}_k, w_g^{t-1})$ 
9:     end for
10:  end for
11:  Upload  $w_k^t (k \in \mathcal{K})$  to the server
12:   $w_g^t = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} w_k^t$ 
13: end for
    
```

The fixed minority label distribution for k -th client is

$$\mu_k(i) = \begin{cases} 1/(C - C_k) & , i \notin \mathcal{Y}_k \\ 0 & , i \in \mathcal{Y}_k \end{cases}, \quad (8)$$

where C and C_k denote the total number of class labels, the number of majority labels for the k -th client, respectively.

As this method does not require the teacher model, it is named FedLMD-Tf (Teacher-free). In this way, such a lightweight version does not increase computation, communication overhead and privacy risk, and can achieve much better performance via perceiving the label distributions. The detailed training is shown in Alg. 1.

5 EXPERIMENTS

5.1 Experimental Setup

Baselines. The methods we compare focus on the traditional FedAvg [28] and on algorithms that are client-side improved for data heterogeneity problems (FedProx [23], FedCurv [39], SCAFFOLD [15], FedNova [43], FedRS [25], MOON [20], FedNTD [19]). For some experiments, we have selected only a few important methods (FedAvg, FedCurv, FedProx, FedNTD) for comparison. All methods are replicated based on the PyTorch framework(1.10.0+cu113) and experimented on RTX 3090 and Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz.

Datasets. For a fair comparison, we decided to use the same experimental setup as [19]. We used four datasets, MNIST [18], CINIC-10 [5], CIFAR-10 and CIFAR100 [17]. The dataset is sliced using two Non-IID partition strategies respectively: 1) **Sharding** [28]: The data is sliced according to labels, and the sliced data is called a slice. Each slice has the same number of examples, and the degree of data heterogeneity is determined by the number of slices s each client has. We set s to MNIST ($s = 2$), CIFAR-10 ($s = 2, 3, 5, 10$), CIFAR-100 ($s = 10$), and CINIC-10 ($s = 2$). 2) **Latent Dirichlet Allocation (LDA)** [20]: The dataset is sliced by dirichlet sampling, which provides unbalanced labels and unbalanced examples for each client.

And the degree of data heterogeneity of different clients is determined by controlling α . We set α as MNIST ($\alpha = 0.1$), CIFAR-10 ($\alpha = 0.05, 0.1, 0.3, 0.5$), CIFAR-100 ($\alpha = 0.1$), and CINIC-10 ($\alpha = 0.1$).

Implementation. For a fair comparison, we use a network model with two convolution layers followed by max-pooling layers, and two fully-connected layers for all methods. The cross-entropy loss and the SGD optimizer are adopted. The learning rate is set to 0.01 and it decays with a factor of 0.99 at each communication round. The weight decay is set to 1e-5 and the SGD momentum is set to 0.9. The batch size is set to 50. For data augmentation, we employ techniques such as random cropping, random horizontal flipping, and normalization. Note that our default experimental dataset is CIFAR-10 ($\alpha = 0.05$) unless specified.

For the FL task, we set some additional hyperparameters. Referring to the settings of previous studies, we set the number of clients $K = 100$, the number of local training epochs $E = 5$, the communication rounds $T = 200$, and randomly select $|\mathcal{K}| = 10$ clients per round.

In all of the experiments, we conduct a grid search on the parameters of each method to determine the optimal performance. After each communication round, we evaluate the global model on the test dataset and select the best test accuracy as the result display.

5.2 Improvement with Knowledge Distillation

In this subsection, we utilize and improve KD to alleviate the situation that the model is prone to be biased toward majority labels under the label heterogeneity case.

First of all, we briefly compared the change in FedAvg accuracy after applying distillation and the results are shown in Fig. 4. We found that KD can help FedAvg alleviate the label heterogeneity problem. However, the traditional KD treats all labels in the same way, which affects the effectiveness of dark knowledge transfer. Therefore, FedNTD only selects non-target labels for distillation. And our proposed FedLMD goes one step further by masking out the majority labels in the output of the teacher model, *i.e.*, selecting minority labels for distillation. From Tab. 1, we can find that FedLMD has significant superiority.

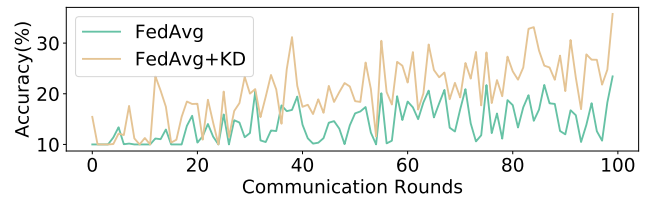


Figure 4: The effect of knowledge distillation in FedAvg on CIFAR-10 ($\alpha = 0.05$).

Moreover, the teacher is often assumed to be a well-pretrained model in KD. But the global model performs poorly at the beginning stage in FL. As shown in Fig. 4, the poor global model as a teacher still improved student performance at the beginning of FL. This observation inspired us to discard the teacher model and use a fixed distribution vector as an unreliable teacher to replace it.

Table 1: The top-1 test accuracy (%) on CIFAR-10 under different distillation methods ($\alpha = 0.05$).

Method	FedAvg	FedAvg+KD	FedNTD	FedLMD
Accuracy	33.02	40.46	47.01	50.45

Further, to better understand the effectiveness of our teacher-free distillation, we use LS ($\mu = 0.1$) on FedAvg which is similar to the teacher-free distillation. In addition, to be fair, we modified FedNTD to a teacher-free version as well, called FedNTD-Tf, for comparison. From the Tab. 2, we find that LS and FedNTD-Tf can alleviate the bias of FL. And when we use teacher-free distillation for minority labels, the FedLMD-Tf is further enhanced by being more focused on preserving the minority label knowledge.

Table 2: The top-1 test accuracy (%) on CIFAR-10 under teacher-free (Tf) distillation ($\alpha = 0.05$). LS: label smoothing.

Method	FedAvg	FedAvg+LS	FedNTD-Tf	FedLMD-Tf
Accuracy	33.02	42.74	41.30	45.08

5.3 Results on Label Heterogeneity

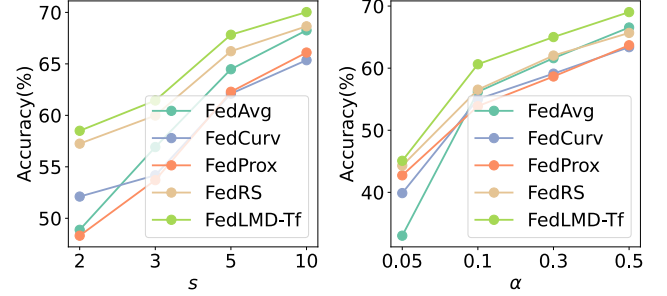
In this subsection, we compare FedLMD and FedLMD-Tf with the previous FL methods comprehensively.

Accuracy and Convergence Speed. We show the results of our experiments with two different strategies of data partition in Tab. 3. The effectiveness of the proposed approach is well illustrated by different datasets and the degree of label heterogeneity. Especially in the case of CIFAR-10 ($\alpha = 0.05$), it is up to 17% improvement over FedAvg. FedLMD outperforms the previous results in the vast most of cases, and the performance improvement becomes more and more obvious as the degree of label heterogeneity increases (α or s keeps decreasing). Even though the results in a few cases are not the best, they are still very close to the SOTA baselines. Moreover, we measure the communication rounds required for different methods to reach the top-1 test accuracy of FedAvg, which is used as the evaluation metric for convergence speed [20]. As shown in in Tab. 3, FedLMD clearly converges faster than the other methods. Specifically, in the experiment on MNIST dataset, it achieves 2.47 times speedup against FedAvg.

We compared the training processes of different methods on the CIFAR-10 dataset. We evaluated their test accuracy on CIFAR-10 under three scenarios: $\alpha = 0.05, 0.3$, and 0.5 . As illustrated in Fig. 6, FedLMD exhibited greater stability during training compared to the other methods in each case scenario.

Comparison with Light Baselines. When FL is deployed on low-power devices, it have to consider the client-side computational costs. Therefore, in such a situation, lightweight FL methods are valuable. Here, we compare the performance of FedLMD-Tf with some previous lightweight approaches on the CIFAR-10 dataset to show its advantage. As shown in Fig. 5, FedLMD-Tf consistently outperforms other methods under various cases without increasing

computational costs. We should additionally note that the size of the vector predefined by FedLMD-Tf on each client depends on the number of class labels C in the FL system.

**Figure 5: Comparison of the accuracy (%) of the method without additional computational cost on two partition strategies Sharding (Left) and LDA (Right) of CIFAR-10.**

5.4 Discussion

Model Architecture. We verify the applicability of the approach with different network architectures, and illustrate the performance with several typical architectures on CIFAR-10 ($\alpha = 0.05$). As shown in Tab. 4, FedLMD works well with these network architectures.

Table 4: The top-1 test accuracy (%) under different networks.

Method	CNN	MobileNet	ResNet-8
FedAvg	33.02	27.84	30.96
FedCurv	39.88	14.10	28.10
FedProx	42.74	30.83	31.32
FedNTD	47.01	30.94	31.41
FedLMD	50.45	31.76	34.60

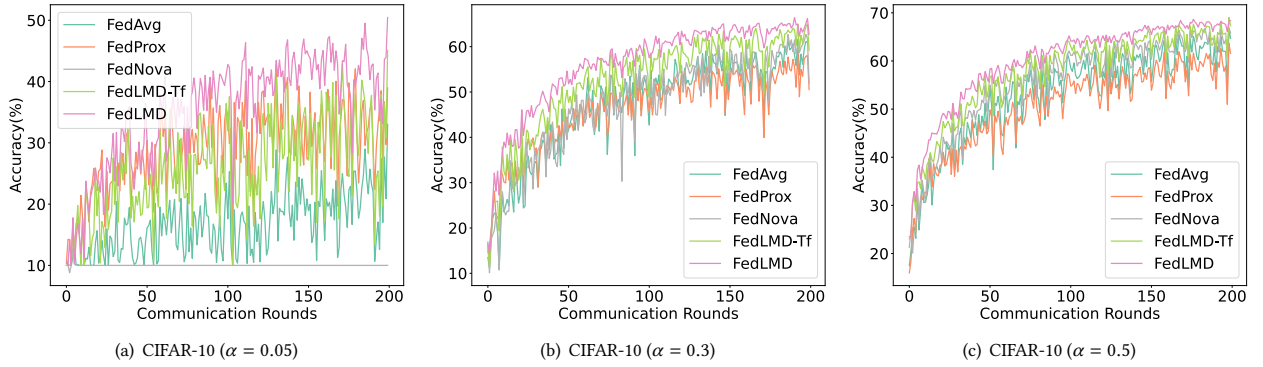
Local Epoch Number. We study the effect of local training epochs on accuracy, and report the results on the left of Fig. 7. As for FedLMD, the enhancement is stable and with excellent performance. While FedLMD-Tf does not perform as well as expected. When $E = 10$, the performance of FedLMD-Tf starts to deteriorate (green line). It may indicate that the client-side model will rely too much on the teacher's performance as E increases, and an unreliable teacher like a fixed distribution vector will limit the optimization of the client-side model with too many local epochs. It is worth pointing out that a larger E will also increase the computational overhead of the client and not all scenarios are better with a larger E [28].

Number of Uploaded Clients. Another point worth discussing in the FL is the number of uploaded clients per communication round. As shown in the right of Fig. 7, the optimal accuracy of each method increases with the number of participating clients. It can be found that FedLMD can achieve good results without having too many models for aggregation, which may be due to its ability to effectively preserve the knowledge of minority labels with a

Table 3: The top-1 test accuracy (%) on MNIST, CIFAR-10, CIFAR-100, and CINIC-10. The values in the parentheses are the speedup of the approach computed against FedAvg. If *Failed* is displayed in the parentheses, the method cannot be converged.

The Non-IID Partition Strategy: Sharding							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$s = 2$	$s = 3$	$s = 5$	$s = 10$		
FedAvg	85.41 (1.00×)	48.88 (1.00×)	56.92 (1.00×)	64.48 (1.00×)	68.26 (1.00×)	26.97 (1.00×)	50.66 (1.00×)
FedCurv	85.08 (1.00×)	52.11 (1.18×)	54.18 (1.00×)	62.10 (1.00×)	65.36 (1.00×)	24.56 (1.00×)	49.52 (1.00×)
FedProx	83.11 (1.00×)	48.31 (1.00×)	53.71 (1.00×)	62.29 (1.00×)	66.10 (1.00×)	26.88 (1.00×)	48.51 (1.00×)
FedNova	85.34 (1.00×)	50.69 (1.07×)	57.98 (1.03×)	65.28 (1.17×)	68.64 (1.09×)	29.11 (1.49×)	49.12 (1.00×)
SCAFFOLD	86.13 (1.41×)	54.62 (1.44×)	40.73 (1.00×)	67.25 (1.59×)	70.79 (1.46×)	31.92 (1.79×)	52.89 (1.54×)
MOON	85.25 (1.00×)	48.40 (1.00×)	57.01 (1.27×)	64.34 (1.00×)	68.36 (1.07×)	27.20 (1.02×)	49.86 (1.00×)
FedRS	85.54 (1.15×)	57.25 (2.60×)	59.98 (1.69×)	66.23 (1.41×)	68.65 (1.07×)	30.24 (1.82×)	51.99 (1.17×)
FedNTD	87.76 (1.98×)	60.03 (3.28×)	61.65 (1.79×)	68.08 (2.02×)	70.06 (1.46×)	32.27 (2.13×)	54.14 (1.69×)
FedLMD	88.48 (2.02×)	60.76 (3.64×)	62.44 (2.04×)	69.20 (2.02×)	70.32 (1.52×)	32.34 (2.30×)	54.13 (2.02×)

The Non-IID Partition Strategy: LDA							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$		
FedAvg	85.19 (1.00×)	33.02 (1.00×)	56.19 (1.00×)	61.61 (1.00×)	66.55 (1.00×)	31.36 (1.00×)	55.64 (1.00×)
FedCurv	84.76 (1.00×)	39.88 (2.35×)	55.00 (1.00×)	59.12 (1.00×)	63.38 (1.00×)	29.65 (1.00×)	54.52 (1.00×)
FedProx	82.43 (1.00×)	42.74 (3.33×)	53.88 (1.00×)	58.66 (1.00×)	63.69 (1.00×)	28.44 (1.00×)	54.06 (1.00×)
FedNova	77.07 (1.00×)	12.71 (<i>Failed</i>)	41.86 (1.00×)	62.70 (1.07×)	66.89 (1.09×)	32.78 (1.33×)	34.67 (1.00×)
SCAFFOLD	81.75 (1.00×)	12.34 (<i>Failed</i>)	28.18 (1.00×)	63.74 (1.24×)	68.07 (1.20×)	34.69 (1.53×)	25.19 (1.00×)
MOON	85.63 (1.24×)	34.54 (1.00×)	56.70 (1.03×)	62.21 (1.07×)	66.43 (1.00×)	31.49 (1.01×)	55.80 (1.04×)
FedRS	85.37 (1.22×)	44.30 (5.00×)	56.54 (1.03×)	62.03 (1.11×)	65.68 (1.00×)	31.51 (1.02×)	58.21 (1.40×)
FedNTD	87.66 (1.60×)	47.01 (5.13×)	60.69 (1.68×)	65.31 (1.71×)	68.10 (1.36×)	33.75 (1.48×)	57.66 (1.57×)
FedLMD	88.61 (2.47×)	50.45 (5.26×)	61.32 (1.77×)	66.43 (2.04×)	68.67 (1.39×)	33.72 (1.46×)	57.73 (1.57×)

**Figure 6: The top-1 test accuracy (%) of different approaches on CIFAR-10 dataset under different communication rounds.**

small number of clients aggregated. As for FedLMD-Tf, it performs similarly to FedAvg when the number of uploaded clients is low. While, when the number of clients increases to 20, FedLMD-Tf has surpassed SOTA baselines with additional computing resources

(such as FedNTD). And when the number of clients is 50, it is already quite close to the teacher version (FedLMD).

Combination with Other FL Methods. We consider the combination of FedLMD and other FL methods for improvement. Here, we

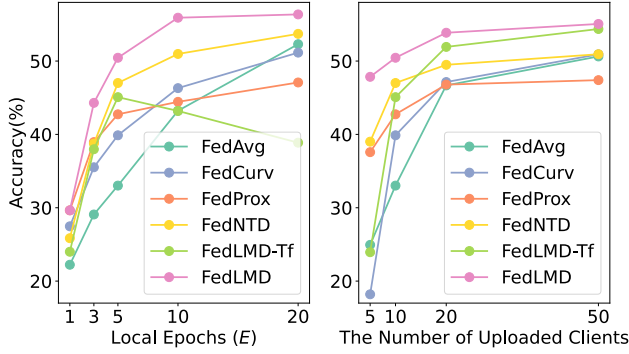


Figure 7: The top-1 test accuracy (%) with different numbers of local epochs (Left) and the uploaded different number of client models (Right).

select two representative methods, FedProx [23] and FedAvgM [35]. FedProx constrains the optimization of the local model from the perspective of model parameters. FedAvgM is based on the Adam optimization algorithm and incorporates a momentum parameter on top of FedAvg. By combining the previous global model parameters with the current aggregated global parameters, it updates the global parameters. As shown in Tab. 5, the combination of FedLMD and FedAvgM performs better than FedLMD on CIFAR-10 ($\alpha = 0.5$), which indicates that the combination of FedLMD and FedAvgM can be applied simultaneously when the label heterogeneity is not very high. Due to the fact that both FedLMD and FedProx are optimization methods with parameter constraints, their optimization trajectories may clash and compromise system performance.

Table 5: The top-1 test accuracy (%) under the combination of FedLMD and other federated learning methods.

Method	FedLMD	+Prox	+AvgM	+Prox+AvgM
Accuracy	68.67	65.42	71.50	70.33

Switching from FedLMD-Tf to FedLMD. As stated in Sec. 4.3, the difference between FedLMD-Tf and FedLMD lies in whether the teacher is used or not. FedLMD-Tf is computationally efficient without a teacher, while FedLMD has a high performance with the teacher. We consider performing FedLMD-Tf first and then switching to FedLMD later since the global model is not a good teacher in beginning. As shown in Fig. 8, we show that the optimization objective improves the performance of the method under different communication rounds of switching from FedLMD-Tf to FedLMD on CIFAR-10 ($\alpha = 0.05$). When the switching round is 200, the method becomes FedLMD-Tf, and when the switching round is 0, it is FedLMD. According to Fig. 8, the performance can be improved by earlier turn switching, which is also accompanied by an increase in computational cost. With such improvements, we can select which round to switch according to actual training situation for balancing between performance and computation.

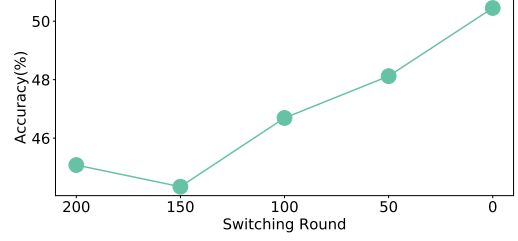


Figure 8: The top-1 test accuracy (%) when switching from FedLMD-Tf to FedLMD after varying rounds.

5.5 Hyperparameters Analysis

Fig. 9 shows the performance of the proposed approach under different hyperparameters. FedLMD achieves excellent performance in most cases, which shows its robustness to the choice of hyperparameters. And for FedLMD-Tf, it suffers from severe performance degradation at higher β . This is mainly due to an unreliable teacher constraining the optimization of the local model. For the temperature τ , a higher value leads to a better performance of FedLMD-Tf, which indicates that a smoother output of the local model is conducive to knowledge retention via teacher-free distillation.

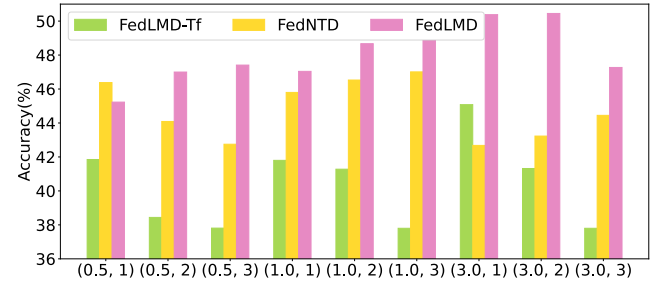


Figure 9: The top-1 test accuracy (%) with different distillation methods under different hyperparameters (τ, β) settings on CIFAR-10 ($\alpha = 0.05$).

6 CONCLUSION

In this paper, we propose FedLMD solve the challenge of label distribution skew in data heterogeneity, which achieves effective and stable FL by retaining knowledge of minority labels. It does not require additional parameters to be uploaded, and thus does not carry additional communication overhead and privacy risk. Our experimental results show that FedLMD is more effective compared to previous methods. Further, considering the limited computational resources on the client-side, we improve it to a teacher-free version. It achieves excellent performance without additional computation. In future work, we will focus on how to apply in larger-scale application scenarios and the optimization solution for other data heterogeneous cases, like the rare labels in the all clients.

Acknowledgements. This work was partially supported by grants from the National Key Research and Development Plan (Grant No. 2020AAA0140001), and Beijing Natural Science Foundation (L192040).

REFERENCES

- [1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. 2021. Federated Learning Based on Dynamic Regularization. In *ICLR*.
- [2] Preston Bukaty. 2019. *The California Consumer Privacy Act (CCPA): An Implementation Guide*. IT Governance Publishing.
- [3] Hongyou Chen and Weilun Chao. 2021. FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning. In *ICLR*.
- [4] Yae Jee Cho, Andre Manoel, Gauri Joshi, Robert Sim, and Dimitrios Dimitriadis. 2022. Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning. In *IJCAI*. 2881–2887.
- [5] Luke N Darlow, Elliot J Crowley, Andreas Antoniou, and Amos J Storkey. 2018. Cinc-10 is not imagenet or cifar-10. *arXiv:1810.03505* (2018).
- [6] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyang Wu, Terrence Chen, David S. Doermann, and Arun Innanje. 2021. Ensemble Attention Distillation for Privacy-Preserving Federated Learning. In *ICCV*. 15056–15066.
- [7] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor W. Tsang, and Masashi Sugiyama. 2020. SIGUA: Forgetting May Make Learning with Noisy Labels More Robust. In *ICML*. 4006–4016.
- [8] Yuting He, Yiqiang Chen, Xiaodong Yang, Hanchao Yu, Yi-Hua Huang, and Yang Gu. 2022. Learning Critically: Selective Self-Distillation in Federated Learning on Non-IID Data. *IEEE Trans. Big Data* (2022).
- [9] Yuting He, Yiqiang Chen, Xiaodong Yang, Yingwei Zhang, and Bixiao Zeng. 2022. Class-Wise Adaptive Self Distillation for Federated Learning on Non-IID Data (Student Abstract). In *AAAI*. 12967–12968.
- [10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In *NeurIPS Workshop*.
- [11] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *arXiv:1909.06335* (2019).
- [12] Ziheng Hu, Hongtao Xie, Lingyun Yu, Xingyu Gao, Zhihua Shang, and Yongdong Zhang. 2022. Dynamic-Aware Federated Learning for Face Forgery Video Detection. *ACM Trans. Intell. Syst. Technol.* 13, 4 (2022).
- [13] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, et al. 2018. Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data. In *NeurIPS Workshop*.
- [14] Peter Kairouz, H Brendan McMahan, Brendan Avent, et al. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14, 1–2 (2021), 1–210.
- [15] Sai Praneeeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *ICML*. 5132–5143.
- [16] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. 2019. NLNL: Negative Learning for Noisy Labels. In *ICCV*. 101–110.
- [17] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto* (2009).
- [18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
- [19] Gihun Lee, Minchan Jeong, Yongjin Shin, Sangmin Bae, and Se-Young Yun. 2022. Preservation of Global Knowledge by Not-True Distillation in Federated Learning. In *NeurIPS*. 38461–38474.
- [20] Qianbin Li, Bingsheng He, and Dawn Song. 2021. Model-Contrastive Federated Learning. In *CVPR*. 10713–10722.
- [21] Qiusi Li, Wenwu Zhu, Chao Wu, Xinglin Pan, Fan Yang, Yuezhi Zhou, and Yaoxue Zhang. 2020. InvisibleFL: federated learning over non-informative intermediate updates against multimedia privacy leakages. In *ACM Multimedia*. 753–762.
- [22] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (2020), 50–60.
- [23] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *MLSys*. 429–450.
- [24] Wenqi Li, Fausto Milletari, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. 2019. Privacy-Preserving Federated Brain Tumour Segmentation. In *MLMI*. 133–141.
- [25] Xin-Chun Li and De-Chuan Zhan. 2021. FedRS: Federated Learning with Restricted Softmax for Label Distribution Non-IID Data. In *SIGKDD*. 995–1005.
- [26] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *NeurIPS*. 2351–2363.
- [27] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. 2021. FedDG: Federated Domain Generalization on Medical Image Segmentation via Episodic Learning in Continuous Frequency Space. *CVPR* (2021).
- [28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, Vol. 54. 1273–1282.
- [29] Virraji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantaha, and Gautam Srivastava. 2021. A survey on security and privacy of federated learning. *Future Generation Computer Systems* 115 (2021), 619–640.
- [30] Dinh C. Nguyen, Quoc-Viet Pham, Pubudu N. Pathirana, Ming Ding, Aruna Seneviratne, Zihui Lin, Octavia A. Dobre, and Won-Joo Hwang. 2023. Federated Learning for Smart Healthcare: A Survey. *ACM Comput. Surv.* 55, 3 (2023), 60:1–60:37.
- [31] Wanning Pan and Lichao Sun. 2021. Local-Global Knowledge Distillation in Heterogeneous Federated Learning with Non-IID Data. *arXiv:2107.00051* (2021).
- [32] Albrecht Jan Philipp. 2016. How the GDPR will change the world. *European Data Protection Law Review* 2, 3 (2016), 287.
- [33] Fan Qi, Zixin Zhang, Xianshan Yang, Huaiwen Zhang, and Changsheng Xu. 2022. Feeling Without Sharing: A Federated Video Emotion Recognition Framework Via Privacy-Agnostic Hybrid Aggregation. In *ACM Multimedia* (Lisboa, Portugal) (MM '22). 151–160.
- [34] Fan Qi, Zixin Zhang, Xianshan Yang, Huaiwen Zhang, and Changsheng Xu. 2022. Feeling Without Sharing: A Federated Video Emotion Recognition Framework Via Privacy-Agnostic Hybrid Aggregation. In *ACM Multimedia*. 151–160.
- [35] Samuel W. Remedios, John A. Butman, Bennett A. Landman, and Dzong L. Pham. 2020. Federated Gradient Averaging for Multi-Site Training with Momentum-Based Optimizers. In *MICCAI Workshop (Lecture Notes in Computer Science, Vol. 12444)*. 170–180.
- [36] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. 2021. FedAUX: Leveraging Unlabeled Auxiliary Data in Federated Learning. *IEEE TNNLS* (2021), 1–13.
- [37] Osama Shahid, Seyedamin Pouriyeh, Reza M Parizi, Quan Z Sheng, Gautam Srivastava, and Liang Zhao. 2021. Communication Efficiency in Federated Learning: Achievements and Challenges. *arXiv:2107.10996* (2021).
- [38] Haizhou Shi, Youcai Zhang, Zijin Shen, Siliang Tang, Yaqian Li, Yandong Guo, and Yuetong Zhuang. 2021. Federated Self-Supervised Contrastive Learning via Ensemble Similarity Distillation. *arXiv:2109.14611* (2021).
- [39] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. 2019. Overcoming forgetting in federated learning on non-iid data. *arXiv:1910.07796* (2019).
- [40] Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuntao Xie, and Weijian Sun. 2020. FedED: Federated Learning via Ensemble Distillation for Medical Relation Extraction. In *EMNLP*. 2118–2128.
- [41] Akihito Taya, Takayuki Nishio, Masahiro Morikura, and Koji Yamamoto. 2021. Decentralized and Model-Free Federated Learning: Consensus-Based Distillation in Function Space. *arXiv:2104.00352* (2021).
- [42] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, et al. 2020. Federated Learning with Matched Averaging. In *ICLR*.
- [43] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 7611–7623.
- [44] Xueyu Wu, Xin Yao, and Cho-Li Wang. 2021. FedSCR: Structure-Based Communication Reduction for Federated Learning. *IEEE Trans. Parallel Distributed Syst.* 32, 7 (2021), 1565–1577.
- [45] Chencheng Xu, Zhiwei Hong, Minlie Huang, and Tao Jiang. 2022. Acceleration of Federated Learning with Alleviated Forgetting in Local Training. *arXiv:2203.02645* (2022).
- [46] Dezhong Yao, Wanning Pan, Yutong Dai, Yao Wan, Xiaofeng Ding, Hai Jin, Zheng Xu, and Lichao Sun. 2021. Local-Global Knowledge Distillation in Heterogeneous Federated Learning with Non-IID Data. *arXiv:2107.00051* (2021).
- [47] Felix X. Yu, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2020. Federated Learning with Only Positive Labels. In *ICML*. 10946–10956.
- [48] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting Knowledge Distillation via Label Smoothing Regularization. In *CVPR*. 3902–3910.
- [49] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. 2022. Fine-tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning. In *CVPR*. 10164–10173.
- [50] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. In *ICML*. 12878–12889.
- [51] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. 2021. Joint Optimization in Edge-Cloud Continuum for Federated Unsupervised Person Re-identification. In *ACM Multimedia*. 433–441.
- [52] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. 2020. Performance Optimization of Federated Person Re-Identification via Benchmark Analysis. In *ACM Multimedia*. 955–963.

Appendix

A EXPERIMENTAL SETUPS

Based on the LDA partition strategy mentioned in the paper, we present the label distribution of CIFAR-10 for 100 clients under different α ($\alpha = 0.05, 0.3, 0.5$) in Fig. 10. The results show that the smaller the value of α , the higher the degree of label heterogeneity.

For the hyperparameter setting of the different methods, for a fair comparison, we use a grid search to obtain the best value of hyperparameters in the different cases for all methods. The detailed used values can be seen in Tab. 6.

B CLASSIFICATION ACCURACY WITH STANDARD DEVIATION

In this section, to ensure the reproducibility of the results under label heterogeneity, we run the main experiments with different random seeds and report the standard deviation values of accuracy. As shown in Tab. 7, the results are generally consistent with the findings in the article.

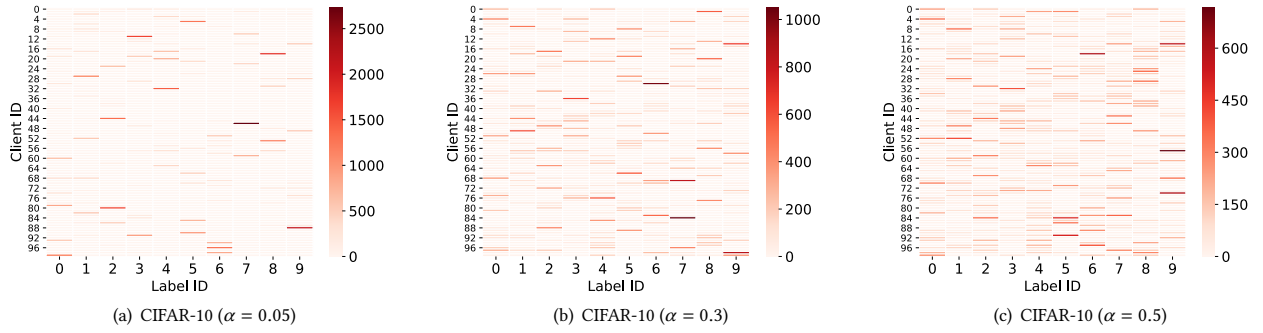


Figure 10: The data distribution of each client for using Non-IID data partitioning. The color bars indicate the number of examples. Each rectangle indicates the number of data examples of a specific class in a party.

Table 6: The hyperparameters used for different methods on MNIST, CIFAR-10, CIFAR-100, and CINIC-10.

The Non-IID Partition Strategy: Sharding							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$s = 2$	$s = 3$	$s = 5$	$s = 10$		
FedCurv	$s = 500, \lambda = 2$	$s = 250, \lambda = 2$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 2$
FedProx	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$
MOON	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 0.5$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$
FedRS	$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.9$
FedNTD	$\beta = 1, \tau = 1$	$\beta = 3, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 3, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 3, \tau = 1$	$\beta = 1, \tau = 1$
FedLMD	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 2, \tau = 3$	$\beta = 1, \tau = 1$
The Non-IID Partition Strategy: LDA							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$		
FedCurv	$s = 500, \lambda = 1$	$s = 250, \lambda = 2$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$	$s = 500, \lambda = 1$
FedProx	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$
MOON	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 0.5$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$	$\mu = 0.1, \tau = 1$
FedRS	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.9$	$\alpha = 0.5$
FedNTD	$\beta = 1, \tau = 1$	$\beta = 3, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 2, \tau = 1$	$\beta = 2, \tau = 1$	$\beta = 3, \tau = 1$	$\beta = 1, \tau = 1$
FedLMD	$\beta = 1, \tau = 1$	$\beta = 2, \tau = 3$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$	$\beta = 1, \tau = 1$

Table 7: The top-1 test accuracy (%) on MNIST, CIFAR-10, CIFAR-100, and CINIC-10. The values in the parentheses are the standard deviation. If Failed is displayed, the method cannot be converged.

The Non-IID Partition Strategy: Sharding							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$s = 2$	$s = 3$	$s = 5$	$s = 10$		
FedAvg	85.09(0.23)	48.75(0.17)	56.88(0.13)	64.85(0.36)	68.76(0.36)	27.30(0.24)	49.38(1.00)
FedCurv	84.79(0.24)	51.11(0.80)	53.48(0.73)	61.87(0.17)	65.35(0.13)	24.98(0.56)	49.35(0.40)
FedProx	82.89(0.27)	49.56(0.90)	54.11(0.56)	62.58(0.23)	66.23(0.16)	26.53(0.30)	48.72(0.16)
FedNova	85.27(0.05)	49.93(0.78)	57.30(0.48)	65.84(0.43)	69.03(0.30)	29.38(0.26)	48.97(0.43)
SCAFFOLD	85.99(0.14)	54.07(0.73)	51.62(7.70)	67.27(0.09)	70.74(0.10)	31.77(0.12)	52.08(0.58)
MOON	84.95(0.27)	48.38(0.10)	56.13(0.75)	64.72(0.34)	68.61(0.19)	27.20(0.04)	49.42(0.45)
FedRS	85.48(0.16)	57.03(0.25)	60.35(0.26)	66.19(0.06)	68.85(0.24)	30.48(0.18)	52.16(0.31)
FedNTD	87.54(0.16)	60.40(0.26)	62.13(0.35)	67.76(1.14)	70.24(0.14)	32.48(0.20)	54.24(0.13)
FedLMD	88.26(0.17)	60.83(0.18)	63.04(0.43)	69.21(0.27)	70.67(0.25)	32.32(0.04)	54.55(0.41)
The Non-IID Partition Strategy: LDA							
Method	MNIST	CIFAR-10				CIFAR-100	CINIC-10
		$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$		
FedAvg	85.44(0.18)	34.08(0.76)	56.40(0.31)	61.57(0.55)	66.48(0.09)	31.66(0.32)	56.00(0.47)
FedCurv	84.78(0.13)	39.95(0.09)	54.87(0.25)	59.22(0.10)	63.17(0.15)	29.73(0.06)	52.48(2.28)
FedProx	82.48(0.07)	42.83(0.10)	54.06(0.35)	58.84(0.31)	63.96(0.37)	28.68(0.24)	53.83(0.54)
FedNova	79.01(2.24)	Failed	41.00(2.21)	62.93(0.30)	67.54(0.51)	33.07(0.29)	36.58(1.36)
SCAFFOLD	80.59(1.65)	Failed	26.87(2.77)	64.00(0.20)	68.32(0.26)	34.73(0.31)	24.35(2.45)
MOON	85.42(0.28)	34.59(0.85)	55.93(0.68)	61.73(0.43)	66.80(0.28)	31.43(0.41)	55.76(0.06)
FedRS	85.42(0.04)	44.47(0.30)	56.65(0.37)	62.07(0.05)	66.36(0.56)	31.84(0.31)	57.80(0.29)
FedNTD	87.13(0.37)	46.72(0.96)	60.26(0.56)	65.73(0.68)	68.28(0.18)	33.97(0.16)	57.13(0.45)
FedLMD	88.06(0.39)	50.18(0.97)	61.25(0.06)	66.64(0.15)	69.15(0.34)	33.79(0.21)	57.66(0.06)