Federated Learning with Hybrid Knowledge Distillations on Long-Tailed Heterogeneous Client Data

Senbin Liu[a,b], Yuanting Zhang[b], Kunhua Zhang[a] and Yi Wang[b,*]

[a]Shenzhen University
[b]Dongguan University of Technology

Abstract. Federated learning (FL) has a great potential in large-scale machine learning applications by training a global model over distributed client data. However, FL deployed in real-world applications often incur collaboration bias and unstable convergence with inconsistent local predictions, resulting in poor modelling performance on heterogeneous and long-tailed client data distributions. In this paper, we reconsider heterogeneous FL in a two-stage learning paradigm where representation learning and classifier re-training are separated to incorporate different sampling schemes. This allows us to deal with the dilemma of obtaining more generalizable features and fine tuning a biased classifier building on client model aggregations. Specifically, we propose a novel hybrid knowledge distillation scheme, called FedHyb, to facilitate the two-stage learning. From the view of knowledge transfer, we show that FedHyb enables several desirable properties in the global feature space and optimization with fine-tuning, thus achieving better test accuracy and convergence speed, especially with a higher level of data heterogeneity and an increasing number of distributed clients. FedHyb does not require any information exchange between clients preventing privacy leakage, and is more robust under poisoning attacks comparing with other FL methods designed on heterogeneous data.

1 Introduction

Federated learning (FL) enables multiple participants to collectively train a common global model by deploying local models at the side of clients without uploading their private data [18]. Despite success in a homogeneous setting, the distributed learning framework still faces challenges in real-world applications where the differences between client participants can impose large effects on model aggregation and federated optimization, leading to significant performance deterioration of FL [29]. For example, in visual recognition applications, real datasets often follow a *long-tailed* distribution where a small portion of classes have a dominant number of instances in the training set comparing to those from the other classes [33]. Figure 1 demonstrates the class-imbalanced distribution with a different degree of data heterogeneity controlled by the Dirichlet distribution coefficient α by sampling the CIFAR-10 dataset over ten image classes across multiple clients. Because each client model is updated on its own data of different tail classes, FL with a long-tailed client distribution tends to incur collaboration bias and unstable convergence. This can result in poor performance of the local models due to local overfitting and a large variance of the aggregated global model [8].

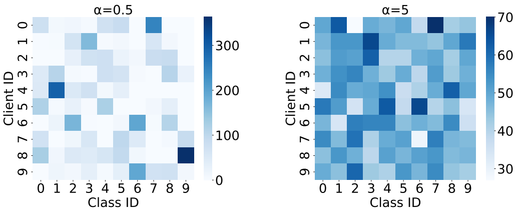


Figure 1. Imbalanced data distribution with a different degree of data heterogeneity over class categories and clients on CIFAR-10.

To resolve the problem of data heterogeneity with FL, most of the existing work adopt one or more of the following strategies: 1) to alleviate local overfitting by making local objectives consistent with the generic global performance [13, 12, 21], 2) to reduce variance of the global model by optimizing server aggregation [15] or participating client clustering/selection [1, 22, 25], and 3) to improve model generalization by combining FL with other methods such as data augmentation [4] and meta-learning [17]. Nevertheless, how to improve communication efficiency and attack robustness while conquering heterogeneity and avoiding privacy leakage remains a key challenge with FL [29].

On the other hand, deep long-tailed learning methods have made remarkable progress in recent years [33]. In particular, a *two-stage* learning paradigm was proposed by decoupling representation learning and classifier training on heterogeneous and long-tailed data [30]. The study finds that 1) *instance-balanced* (natural) sampling learns the best and most generalizable representations and 2) re-adjusting the classifier with *class-balanced* sampling leads to significant performance improvement in long-tailed recognition. In [21], an empirical study on a FedAvg model and client datasets showed that the biased classifier is the primary factor degrading the performance of heterogeneous FL. Accordingly, it proposed to re-train the global model based on FedAvg [18] with a set of balanced features, called federated features, whose gradients are made close to those of real data during the client updates. However, the class-balanced federated features learned at the server do not follow natural sampling. Thus, the resulting global representations are less generalizable when they are applied to other datasets different from the client data distribution.

In this paper, we propose to coordinate the above seemingly contradicting sampling strategies in the two-stage learning paradigm for FL on heterogeneous and long-tailed client data. This is facilitated by a hybrid knowledge distillation scheme, namely FedHyb, as illustrated in Figure 2. Specifically, we adopt the conventional instance-

[*] Corresponding Author. Email: wangyi@dgut.edu.cn

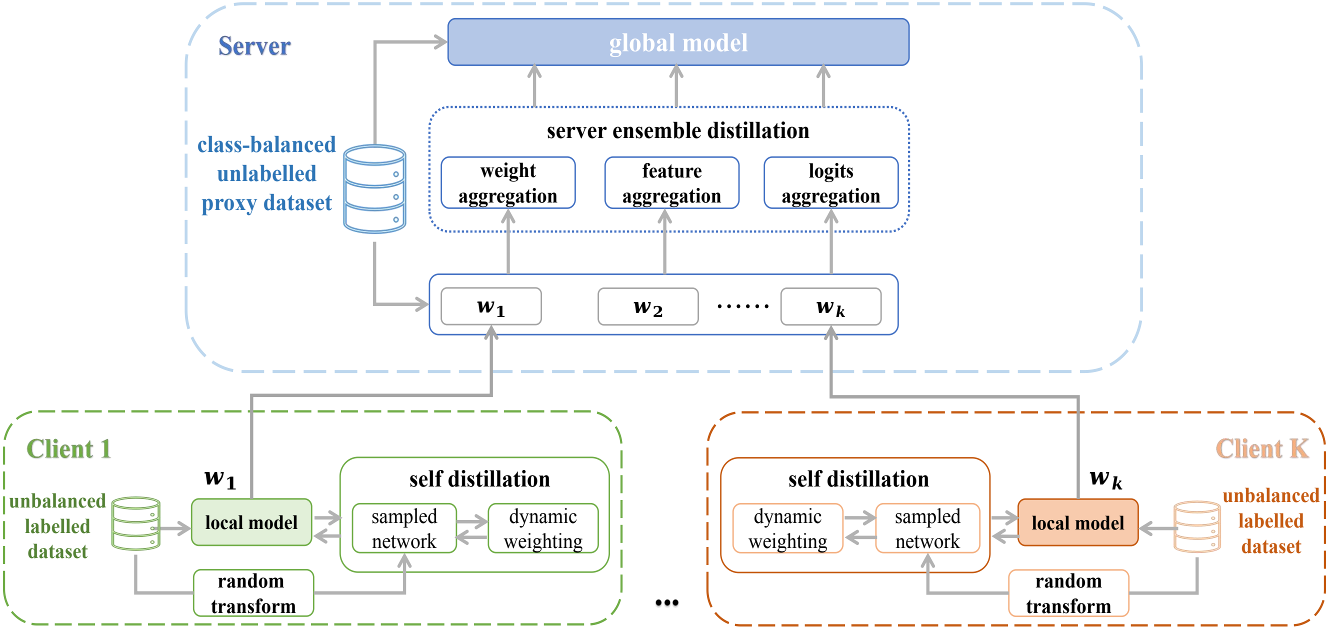


Figure 2. The proposed FedHyb scheme for two-stage learning on heterogeneous and long-tailed client data.

based sampling with local labelled dataset at individual client sides. The updated client models are sent to the server to obtain aggregated feature representations which inherently follow the real-data natural distribution over the clients. Instead of building the global model directly on weight aggregation as in FedAvg, we propose a server ensemble distillation scheme that transfers the aggregated client information to the global model more comprehensively at three different levels. The server distillation process is guided by a class-balanced *unlabelled* dataset. The auxiliary dataset can be either from a third party or even synthetically generated, where the sample is not in the client data distribution, for supervising the ensemble knowledge transfer. In this way, we are able to learn more generalized feature representations from model aggregation while re-training the classifier using a set of balanced data and thus achieving better final test accuracy.

The contributions of this paper are:

- We design a two-stage learning scheme called FedHyb that consists of client-side self distillation to alleviate local over-fitting and server ensemble distillation to improve model aggregation and generalization for FL on heterogeneous and long-tailed client data. The proposed FedHyb scheme does not require any information exchange between clients nor global data distribution for federated optimization, and thus preventing privacy leakage.
- We show that FedHyb enables the desirable properties of representation learning and flatter loss landscapes for fine-tuning the aggregation model, both of which help to improve the global model accuracy with an increasing number of clients at different degrees of data heterogeneity.
- We provide theoretical convergence analysis and empirical results to demonstrate the communication efficiency of FedHyb. In general, compared with existing SOTA methods of heterogeneous FL, FedHyb requires less communication rounds to reach a target accuracy and yields a more stable and robust training performance.

2 Related Work

Previous work showed that model aggregation of homogeneous FL by FedAvg has a poor performance in terms of both test accuracy and communication cost on non-independent identically distributed (Non-IID) data due to inconsistent update optimization directions of participants [14]. Since then, different methods were proposed to alleviate various problems caused by data (statistical) heterogeneity for FL in more realistic settings. Instead of categorizing on strategies, a recent survey reviewed current heterogeneous FL methods by introducing a new taxonomy at three different levels [29]. The data-level methods introduce operations to smooth the statistical heterogeneity of local data across clients. This category includes private data processing such as data augmentation [4, 34] and external data utilization [31]. Model-level methods tend to operate at the model level, which aim to learn a local model for each client that adapts to its private data distribution while learning the global information. These include adding regularization (e.g., FedProx [13]), incorporating contrastive learning (e.g., MOON [12]), meta learning [17], improving consistency [15, 21], and sharing partial structures [6]. Server-level methods require server participation, such as participating client selection [25, 22] or client clustering [1].

Recently, there is a trend to combine FL with knowledge distillation, known as *federated distillation* (FD), which can leverage external data sources and knowledge transfer to improve FL performance. FD belongs to both data-level and model-level methods [29]. There are mainly two ways of deploying FD: 1) by client distillation and 2) by server distillation.

In client distillation such as [10, 34, 5], each client obtains the averaged soft predictions from all clients to constrain local updates and prevent falling into local optimality. For example, FedMD [10] implements client-side communication by leveraging FD and transfer learning. It calculates a global consensus through a common averaging strategy from all client models. Such client distillation methods often require a public dataset be provided to all clients, and the performance heavily depends on the public data quality. To resolve this limitation, FedHKD [5] computes mean representations and the corresponding mean soft predictions for the data classes as "hyper-knowledge". The globally aggregated hyper-knowledge instead of public dataset is used by clients in the subsequent training epoch. However, it still requires data exchange between clients that may cause privacy leakage [27], and is prone to poisoning attacks by malicious clients [1]. In our design of FedHyb, the self-distillation scheme avoids client-side communication.

In server distillation, the server typically aggregates client models, averages the clients' soft predictions, and uses an auxiliary dataset to fine-tune the global model [32]. To alleviate the reliance on public data for distillation, FedDF [16] utilizes unlabeled or generated data to construct the auxiliary dataset. Similarly, FedGen [34] is also unsupervised learning where each client directly regulates the local updates using unlabeled samples generated on-the-fly on the server, whereas FedFTG [32] trains a conditional generator to fit the input space of a local model and uses it to generate pseudo data. However, these methods have excessive computation costs in training the generator and using KL divergence to learn the global knowledge. DaFKD [26] takes into account the domain knowledge for training local models and endows the local models with different importance to learn soft predictions across clients. Nevertheless, the SOTA method still relies on the ensemble of local predictors for distillation, making it sensitive to misleading and ambiguous knowledge injected by poorly performed local model(s). Sharing soft predictions only exacerbates this problem [22]. In our design of FedHyb, the server ensemble distillation scheme incorporates rich information from data representations for global model optimization.

3 Proposed Methodology

Consider FL with K clients. Client k, for $k = 1, 2, ..., K$, can only access its own private data with labels $\mathrm{D}_k = (\mathbf{x}, \mathbf{y})$ at the local side. Denote the client data volume by $N_k = |\mathrm{D}_k|$. The participating clients perform local training on D_k with the cross-entropy (CE) loss

$$\min_{\omega} F_k(\omega; \mathbf{x}, \mathbf{y}) := \frac{1}{N_k} \sum_{i=1}^{N_k} \mathcal{L}_{\mathrm{CE}} \left(\omega; x_i^k, y_i^k \right) \tag{1}$$

and send to the server the local client model parameters denoted by $\omega_1, \omega_2, ..., \omega_K$. The server then performs a simple model aggregation by *weight aggregation* to obtain the global model parameters [18]

$$\Omega = \sum_{k=1}^{K} p_k \omega_k \tag{2}$$

where $p_k = N_k / |\mathrm{D}|$ is a weighing factor of client k's data volume with respect to all data volumes $\mathrm{D} = \cup \{\mathrm{D}_k\}_{k=1}^{K}$ for $k = 1, 2, ..., K$. The server distributes the global model parameters Ω back to the clients. The process repeats for T rounds till convergence. Note that, in homogeneous FL, the simple weight aggregation in (2) is effectively equivalent to minimizing the averaged CE loss over the client models.

3.1 Client Self-Distillation

Previous studies showed that regularization can limit the local updates of the client, thereby mitigating the impact of data heterogeneity [13, 12, 28]. Existing work performs local logits distillation but requires information exchange between clients [10]. If a malicious client is involved, the privacy of other clients may be leaked through the collaborative learning process. To resolve this problem, we perform client self distillation which is privacy preserving without the need of information exchange nor a public dataset shared between the clients.

In the self-distillation process, we firstly perform random transforms, including scaling and rotations, on the local data samples to obtain the *client distillation data*. This is denoted by $R_m(\mathbf{x})$ for $m = 1, 2, ..., M$, where M is the number of random transforms for each $\mathbf{x} \in \mathrm{D}_k$ with *instance-based* resampling. On the other hand, we also sample M sub-networks with different network fraction width from the original network, denoted by $S_m(\omega)$ for $m = 1, 2, ..., M$. We then use the accumulated distillation loss of M sub-networks to limit the local updates and alleviate local over-fitting. Specifically, we introduce the following regularization term on the conventional CE loss function for regularizing local model updates at Client k:

$$\mathcal{L}_{\mathrm{k}}(\omega; \mathbf{x}) = \sum_{m=1}^{M} \mathrm{KL} \left(Q_k \left(\omega; \mathbf{x} \right) \| Q_k \left(S_m(\omega); R_m(\mathbf{x}) \right) \right) \tag{3}$$

where the softmax output of the m-th slimmed sub-network trained on the corresponding client distillation dataset, denoted by $Q_k \left(S_m(\omega); R_m(\mathbf{x}) \right)$, is aligned to the original softmax output $Q_k \left(\omega; \mathbf{x} \right)$ of client k's local model on their KL divergence. The proposed client distillation scheme involves sampling of multiple sub-networks with different network fraction width which is fine-tuned with client distillation data randomly generated by affine transforms of the local dataset with instance-based resampling. It serves to enrich the local semantic information of learning without privacy leakage to other clients.

To account for different learning abilities, we further design a *dynamic weighing* scheme to adjust the individual contribution of each sub-network in the distillation process based on their predictive performance. Specifically, we calculate the importance factor of each sub-network by sigmoid function

$$\beta_m(\omega; \mathbf{x}, \mathbf{y}) = \frac{\exp \left(-\mathcal{L}_{\mathrm{CE}} \left(S_m(\omega); R_m(\mathbf{x}), \mathbf{y} \right) \right)}{1 + \exp \left(-\mathcal{L}_{\mathrm{CE}} \left(S_m(\omega); R_m(\mathbf{x}), \mathbf{y} \right) \right)} \tag{4}$$

where $\mathcal{L}_{\mathbf{CE}} \left(S_m(\omega); R_m(\mathbf{x}), \mathbf{y} \right)$ is the CE loss of the m-th slimmed sub-network of client k calculated on $R_m(\mathbf{x})$ with the original labels $\mathbf{y}$. In this way, a sub-network with higher confidence of prediction will contribute more to the client self distillation process.

Combining the above elements, the objective function of a local model at client k is

$$\min_{\omega} F_k(\omega; \mathbf{x}, \mathbf{y}) + a \sum_{m=1}^{M} \beta_m(\omega; \mathbf{x}, \mathbf{y}) \mathcal{L}_{\mathrm{k}}(\omega; \mathbf{x}) \tag{5}$$

where the first part is the conventional CE loss as shown in (1), and the second part is the accumulated knowledge distillation loss based on soft label predictions, as shown in (3), with dynamic weighing for adaptive model regularization. The contributions of the CE loss and the knowledge distillation loss are balanced with a hyper-parameter a. The client self-distillation process is illustrated for two clients in Figure 2.

3.2 *Server Ensemble Distillation*

Recent advances in long tailed learning found that decoupling representation learning from classifier fine-tuning can significantly improve long-tailed class recognition [30]. In particular, retraining the classifier with class-balanced sampling is beneficial in re-adjusting the decision boundaries. Accordingly, we are inspired to leverage FD to facilitate external data utilization. As pointed out in [22], ensemble predictions may be ambiguous and exhibit high entropy when local predictions of the clients are highly inconsistent on heterogeneous data, which results in poor performance by weight aggregation only [14] and harms FD with soft label predictions [29].

We propose to resolve the problem by server ensemble distillation that transfers more comprehensive aggregated information from client updates to the global model as shown in Figure 2. Specifically, we perform three levels of aggregation including weight aggregation, logits aggregation and feature aggregation. Note that the latter two are reconstructed with the uploaded client model parameters $\omega_1, \omega_2, ..., \omega_k$ only and thus can save communication costs. The server ensemble distillation scheme is guided by a *class-balanced* unlabelled proxy dataset, denoted by $\mathrm{D}_S := \hat{\mathbf{x}}$. We show later in Section 4 that this proxy dataset can be either from a third-party source or synthetically generated that is irrelevant to the natural data distribution distributed at client sides.

We first perform weight aggregation with (2) to obtain the global model Ω and then fine-tune Ω with joint logits and feature distillations with D_S. Similar to the knowledge distillation loss defined in (3), the logits knowledge transfer is performed through imposing a regularization term

$$\mathcal{L}_{\text{logits}}(\Omega; \hat{\mathbf{x}}) := \mathrm{KL}\left(\frac{1}{K}\sum_{k=1}^{K} Q\left(\omega_k; \hat{\mathbf{x}}\right) \| Q_{\mathrm{G}}\left(\Omega; \hat{\mathbf{x}}\right)\right) \tag{6}$$

where the first term is the aggregated softmax output of the uploaded client models by logit aggregation, and the second one is that of the global model with current network parameters Ω before fine-tuning.

We also transfer the knowledge of feature extraction by K clients to the server. This is done by minimizing the mean square error (MSE) distance between the data representation output of the server model and that of client models. The feature knowledge transfer is performed through the following regularization term:

$$\mathcal{L}_{\text{feature}}(\Omega; \hat{\mathbf{x}}) := \mathrm{MSE}\left(H_{\mathrm{G}}\left(\Omega; \hat{\mathbf{x}}\right), \frac{1}{K}\sum_{k=1}^{K} H\left(\omega_k; \hat{\mathbf{x}}\right)\right) \tag{7}$$

where the first term in MSE represents the penultimate features extracted by the global model Ω, while the second term is the aggregated penultimate features obtained from the uploaded client models with ω_k for $k = 1, 2, ..., K$ on the server auxiliary data $\hat{\mathbf{x}}$. By combining both types of knowledge transfer, the server ensemble distillation loss is therefore

$$\min_{\Omega} \eta \mathcal{L}_{\text{logits}}(\Omega; \hat{\mathbf{x}}) + \nu \mathcal{L}_{\text{feature}}(\Omega; \hat{\mathbf{x}}) \tag{8}$$

where η and ν are used to adjust the proportion of soft prediction knowledge and representation knowledge. Algorithms 1 and 2 summarize the main procedures of the proposed FedHyb scheme at both the server and client sides.

3.3 *Effects on Feature Learning and Fine-Tuning*

In this section, we show that the proposed ensemble distillation scheme can enable desirable properties of representation and classifier learning that help to improve the model test accuracy. Firstly, we use t-distributed stochastic neighbor embedding (t-SNE) [24] to visualize the implicit data structure of deep features, which converts the high-dimensional Euclidean distances between data points into conditional probabilities that present similarities. The experiments are performed with 500 image examples from four classes in the benchmark CIFAR-10 dataset. Figure 3 plots the t-SNE map of deep features extracted from the global model based on weight aggregation for the case without knowledge distillation (i.e., FedAvg), the case with logits distillation only, and the case with joint knowledge transfer of logits and feature representations as expressed in (8). It shows that FL with the proposed ensemble distillation is able to learn more *compact* and *discriminative* representations in the global feature space which helps to improve the recognition performance.

We also use the method in [11] to visualize and compare the training loss landscapes by the different aggregation models. In the visualization method, the x-axis is the magnitude and the model weights are perturbed by a series of Gaussian noises with varying degrees. Following [11], each noise level is normalized to the l_2 norm of each filter to account for the effects of varying weight amplitudes of different models. In Figure 4, each plot has 10 landscapes using 10 randomly generated directions. It can be seen that adding knowledge distillation can encourage flatter loss landscape, which helps the opti-

Algorithm 1 FedHyb on Server

```
Input: Public unlabeled dataset $\mathrm{D}_S$
Output: Global model $\Omega$
Server executes:
set $t = 0$ and randomly initialize $\Omega^t$
for communication round $t < \mathrm{T}$ do
    let $t \leftarrow t + 1$ and broadcast $\Omega^t$ to $K$ clients
    for client $k \in K$ clients do
        receive $\omega_k^t$ from client $k$
    end for
    update $\Omega^t$ with $\{\omega_k^t\}$ by weight aggregation in (2)
    update $\Omega^t$ with $\mathrm{D}_S$ by ensemble distillation in (8)
end for
return $\Omega^{\mathrm{T}}$
```

Algorithm 2 FedHyb on Client k

```
Input: global model $\Omega^t$ and local dataset $\mathrm{D}_k$
Output: Client $k$ model $\omega_k$
Client $k$ executes:
set $\omega_k = \Omega^t$
update $\omega_k$ with $\mathrm{D}_k$ by client logits distillation in (5)
return $\omega_k$
```

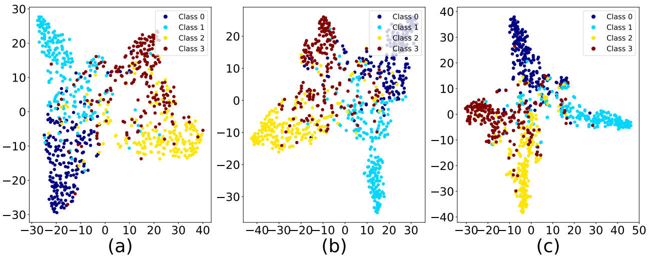


Figure 3. t-SNE visualization of 500 CIFAR-10 image examples for the global model learned (a) without knowledge distillation (i.e., FedAvg), (b) with logits distillation only, and (c) with the proposed ensemble distillation.

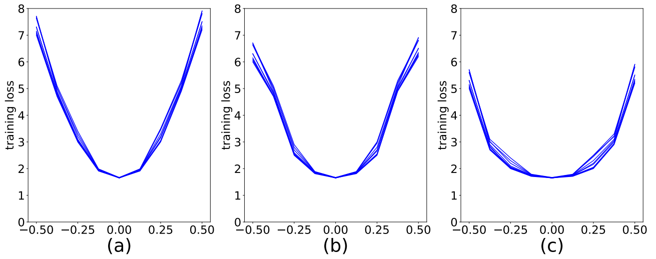


Figure 4. The loss landscapes by fine-tuning the aggregation model (a) without knowledge distillation (i.e., FedAvg), (b) with logits distillation only, and (c) with the proposed ensemble distillation.

mization process in fine-tuning [11]. In particular, the loss landscape by incorporating both logits and feature alignments in the ensemble distillation is most beneficial for fine-tuning the aggregation model.

4 Experiments

4.1 Experiment Setup

Datasets. We perform evaluations on three benchmark datasets of SVHN [20], CIFAR-10 and CIFAR-100 [9]. Similar to existing work, we use Dirichlet distribution Dir(α) on label ratios to generate the Non-IID data distribution among clients, where a smaller α closer to 0 indicates higher data heterogeneity as shown in Figure 1. In this way, we can produce a class-imbalanced D_k of 500 samples with data heterogeneity controlled by α for each client. The test data from the benchmark datasets are used for performance evaluation. We also construct a class-balanced unlabelled proxy dataset D_S on the server where the samples can be from a third-party resource (e.g., Internet[16, 3]) or synthesized with a generator (e.g., GAN[32, 26]). We mark our method trained on the former by FedHyb and that trained on the latter by FedHyb*. Both types of D_S are irrelevant to D_k used on the client sides.

Network Architecture. In this work, we use the same network architecture for the basic backbone on the server and clients. Specifically, we deploy ResNet18 for CIFAR-10 and CIFAR-100, and ShuffleNet-V2 for SVHN, respectively.

Experimental Setting. The batch size of local training is set to 64 and the learning rate is 0.001. The number of clients increases at 10, 20, and 50. The maximum number of communication rounds T is set to 50. The number of local epochs is set to 5. The number of subnetwork is 2. For performance evaluation, we test the proposed FedHyb scheme on the heterogeneous data sets in comparison with eight popular FL methods: FedAvg [18], FedProx [13], MOON [12], FedGen [34], FedMD [10], FedDF [16], FedHKD [5] and DaFKD [26]. In particular, the latter five are the SOTA distillation-based methods. Unless otherwise specified, the comparing methods are run with their default settings reported in the corresponding paper.

4.2 Performance Analysis

Table 1. Model test accuracy at a high data heterogeneity with $\alpha = 0.5$.

Dataset	Scheme	Local Model Accuracy			Global Model Accuracy		
	# clients	10	20	50	10	20	50
SVHN	FedAvg [18]	0.6766	0.7329	0.6544	0.4948	0.6364	0.5658
	FedProx [13]	0.6927	0.6717	0.6991	0.5191	0.6419	0.6139
	Moon [12]	0.6602	0.7085	0.7192	0.4883	0.5536	0.6543
	FedGen [34]	0.5788	0.5658	0.4679	0.3622	0.3421	0.3034
	FedMD [10]	0.8038	0.8086	0.7912	0.6812	0.7344	0.8085
	FedDF [16]	0.7824	0.7953	0.7805	0.6321	0.7053	0.7334
	FedHKD [5]	0.8086	**0.8381**	0.7891	0.6781	0.7357	0.7891
	DaFKD [26]	0.7948	0.8059	0.7798	0.6681	0.7405	0.7794
	FedHKD+DaFKD	0.7812	0.7851	0.7529	0.6617	0.7291	0.7924
	FedHyb	**0.8112**	0.8241	**0.8291**	0.6902	**0.7513**	**0.8104**
	FedHyb*	0.8109	0.8257	0.8275	**0.6941**	0.7508	0.8097
CIFAR-10	FedAvg [18]	0.5950	0.6261	0.5825	0.4741	0.5516	0.3373
	FedProx [13]	0.5981	0.6295	0.6490	0.4793	0.5258	0.5348
	Moon [12]	0.5901	0.6482	0.5513	0.4579	0.5651	0.3514
	FedGen [34]	0.5879	0.6395	0.6533	0.4800	0.5408	0.5651
	FedMD [10]	0.6147	0.6666	0.6533	0.5088	0.5575	0.5714
	FedDF [16]	0.6341	0.6535	0.6543	0.4921	0.5453	0.5213
	FedHKD [5]	0.6254	0.6816	0.6671	0.5213	0.5735	0.5493
	DaFKD [26]	0.6331	0.6748	0.6581	0.5285	0.5681	0.5681
	FedHKD+DaFKD	0.6065	0.6872	0.6694	0.5347	0.5672	0.5617
	FedHyb	0.6591	**0.6993**	**0.6934**	**0.5531**	0.5920	**0.5860**
	FedHyb*	**0.6595**	0.6987	0.6901	0.5514	**0.5973**	0.5829
CIFAR-100	FedAvg [18]	0.2361	0.2625	0.2658	0.2131	0.2748	0.2907
	FedProx [13]	0.2332	0.2814	0.2955	0.2267	0.2708	0.2898
	Moon [12]	0.2353	0.2729	0.2428	0.2141	0.2652	0.1928
	FedGen [34]	0.2393	0.2701	0.2739	0.2176	0.2620	0.2739
	FedMD [10]	0.2681	0.3054	0.3293	0.2323	0.2669	0.2968
	FedDF [16]	0.2642	0.2913	0.3170	0.2154	0.2543	0.2732
	FedHKD [5]	0.2981	**0.3245**	0.3375	0.2286	0.2795	0.2988
	DaFKD [26]	0.2682	0.2978	0.3278	0.2291	0.2857	0.2818
	FedHKD+DaFKD	0.2818	0.3193	0.3324	0.2105	0.2748	0.2901
	FedHyb	**0.3074**	0.3220	**0.3515**	0.2332	0.3012	**0.3263**
	FedHyb*	0.2997	0.3185	0.3426	**0.2431**	**0.3075**	0.3247

Test Accuracy. Table 1 compares the test results of global and local model accuracy on the three benchmark datasets. The data heterogeneity is set to a high level with $\alpha = 0.5$ resulting in a long-tailed distribution. In general, the proposed method of FedHyb is able to achieve the highest accuracy in most cases for the global model at the server side, especially when the number of clients increases from 10 to 50. For example, FedHyb achieves a gain of 11-24% on server accuracy comparing with FedAvg, about 2-3% comparing with the SOTA method of FedHKD with client distillations, about 2-3% comparing with the SOTA method of DaFKD with server distillations. In terms of the client model accuracy, FedHyb still leads the performance in most cases and on a par with FedHKD in the case of 20 clients for SVHN and CIFAR-100. FedHyb also outperforms the combined approach of FedHKD+DaFKD, indicating the advantage of the two-stage learning paradigm over a straightforward combination of client and server distillation methods. Note that the proposed server ensemble distillation scheme trained with different proxy dataset D_S, i.e., FedHyb vs. FedHyb*, perform very closely over all three benchmark datasets. This indicates the role of proxy data in fine-tuning the aggregation model as guiding the alignments rather than recognition in the conventional tasks. In fact, it is the class-balanced proxy dataset that helps to re-adjust the decision boundaries of a biased classifier obtained by aggregating the client updates over heterogeneous data.

Convergence Analysis. The communication bottleneck of FL is the connection between the central server and clients, which is generally slow due to two-round communications, i.e., one broadcast and one aggregation, per iteration of training. Thus, convergence of the global model significantly affects the speed of the overall FL process. In particular, it was demonstrated in [14] that the heterogeneity of training data further slows down the convergence speed. It also provided theoretical guarantees for the convergence results of FedAvg with federated averaging using (2) on Non-IID data. That is

$$\mathbb{E}\left[F\left(\boldsymbol{\Omega}_t\right)\right] - F^* \leq \frac{\kappa}{\gamma + t - 1}\left(\frac{2B}{\mu} + \frac{\mu\gamma}{2}\mathbb{E}\left\|\boldsymbol{\Omega}_1 - \boldsymbol{\Omega}^*\right\|^2\right) \tag{9}$$

where after t rounds the difference expectation between the global

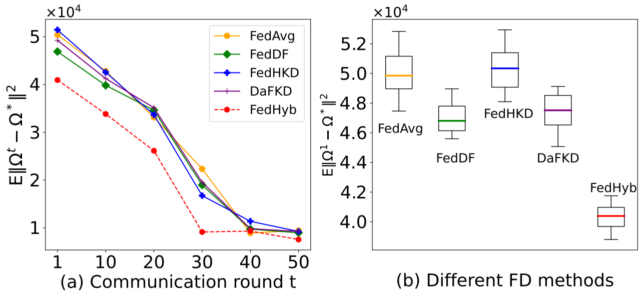


Figure 5. Convergence analysis by the upper bound $\mathbb{E}\|\mathbf{\Omega}_t - \mathbf{\Omega}^*\|^2$.

Table 2. Communication rounds (on average) required to reach a target model accuracy (acc.) on CIFAR-10 with $\alpha = 0.5$.

Scheme	acc.=70%	acc.=80%	acc.=90%	acc.=100%
FedAvg	10.33	22.67	34.33	41.00
FedProx	12.67	22.33	32.00	40.33
Moon	18.67	27.00	35.33	44.67
FedGen	16.00	33.33	38.67	42.33
FedMD	9.33	21.67	27.00	39.67
FedDF	8.67	19.00	28.67	37.33
FedHKD	12.33	23.33	27.67	40.33
DaFKD	9.00	19.67	29.67	38.33
FedHyb	**7.33**	**15.67**	**25.33**	**32.67**

objective and its optimal value, i.e., $\mathbb{E}\left[F\left(\mathbf{\Omega}_t\right)\right] - F^*$, is upper bounded by the l_2-norm distance expectation of the global model and its optimal case in the parameter space when all other problem-related parameters and conditions are determined.

We then plot the upper bound variable $\mathbb{E}\|\mathbf{\Omega}_t - \mathbf{\Omega}^*\|^2$ with respect to the communication round t for the fastest comparing methods in Figure 5(a). It can be seen that the proposed method of FedHyb can significantly reduce the upper bound and thus accelerate the global model convergence at about $t = 30$ in advance. Figure 5(b) further evaluates the stability of network convergence by drawing box-plots of the first-round model discrepancy for the comparing methods. In addition to a remarkably lower expectation of the upper bound variable, the variance of the l_2-norm difference between the global model and the optimal case is also the smallest for FedHyb in the parameter space, indicating a more stable and robust training performance.

Communication Efficiency. As shown above, FedHyb is able to accelerate the global model convergence with better stability and robust training performance. This directly affects the communication speed by improving the FL training efficiency. Table 2 verifies the convergence analysis by comparing the communication rounds between FebHyb and the other methods for the global model to reach a target accuracy on CIFAR-10 with ten clients and $\alpha = 0.5$. It can be seen that FedHyb achieves the best convergence speed in all cases by requiring less communication rounds on average for running three times of the training process.

Data Heterogeneity. We further test the global model accuracy on CIFAR-10 and SVHN with an increasing level of data heterogeneity by varying α. The results are plotted in Figure 6 and Figure 7. As α increases, the data heterogeneity decreases as the client data becomes more evenly distributed over different image classes. It can be seen that almost all methods have their test accuracy improved as the data heterogeneity decreases, indicating a severe impact of the heterogeneity degree on the model performance. In particular, FedHyb outperforms the comparing methods in all cases. For examples, it achieves a significant gain over the baseline method of FedAvg by almost 20% and the SOTA methods of FedHKD and DaFKD by up to 3% when $\alpha = 0.2$, demonstrating the effectiveness of our method under extreme data heterogeneity distribution.

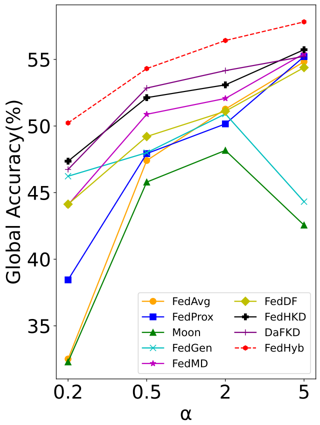


Figure 6. Impact of heterogeneity (on CIFAR-10).

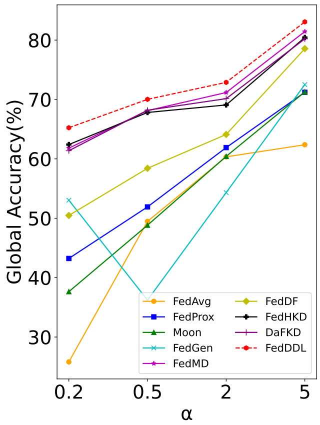


Figure 7. Impact of heterogeneity (on SVHN).

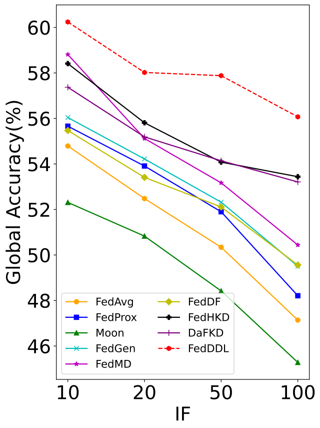


Figure 8. Impact of long-tail (on CIFAR-10).

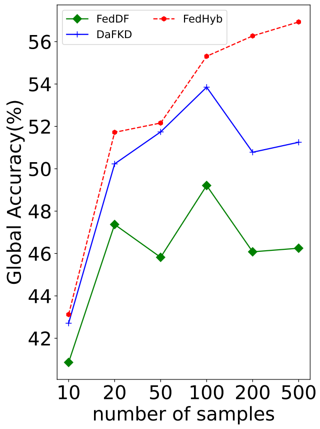
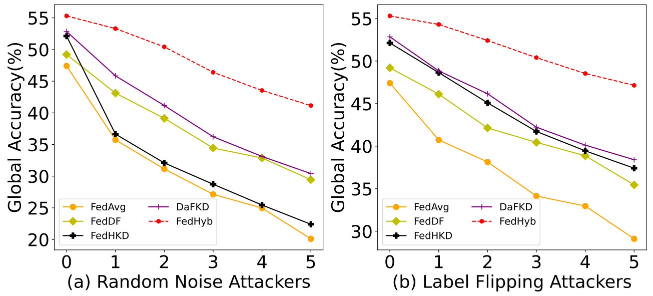


Figure 9. Impact of proxy data size (on CIFAR-10).

Data Long-Tail. In this section, we test our method on a special type of heterogeneous data with long-tailed distributions. Specifically, we generate the long-tailed datasets, namely CIFAR-10-LT, based on the original dataset of CIFAR-10. The degree of long-tailed distribution can be controlled by introducing an imbalance factor (IF) as in [2]. Note that a higher IF value indicates a more imbalanced data distribution with a longer tail as the client data becomes more unevenly distributed over different image classes. In our experiments, we test on four CIFAR-10-LT with different long-tailed heterogeneous data distributions by setting IF=10, 20, 50, 100, respectively. The results are plotted in Figure 8. It can be seen that IF has a significant impact on the model performance for all comparing methods as the test accuracy decreases with an increasing degree of data long-tail. Some method such as MOON [12] can suffer from significant performance degradation under extreme long-tailed distribution, whereas the proposed method of FedHyb is more robust when the IF degree is increased from 10 to 100. In all cases, FedHyb outperforms the other



(a) Random Noise Attackers

(b) Label Flipping Attackers

Figure 10. Model robustness under different poisoning attacks with malicious clients.

Table 3. Model test accuracy on real-world dataset.

Method	FedAvg	FedDF	FedHKD	DaFKD	FedHyb
CheXpert	0.6341	0.6623	0.6759	0.6816	**0.7032**

methods by up to 11%. In particular, the gain of test accuracy is 8% over FedAvg and about 3-4% comparing with the SOTA methods of DaFKD and FedHKD.

Proxy Dataset Size. Figure 9 shows the impact of proxy dataset by varying the number of samples used for server distillation. Note that the class-balanced proxy dataset is irrelevant to the heterogeneous client dataset. We compare the global accuracy of FedHyb with two other server distillation methods, namely FedDF and DaFKD. It can be seen that our method is able to further improve the global model accuracy by increasing the proxy dataset size for fine-tuning. This indicates a significant advantage of FedHyb in the scalability of the server distillation process.

Attack Robustness. We use two attack methods to evaluate the robustness of our method with malicious clients who try to poison FL [19]. In particular, random noise attacks generate perturbations based on Gaussian distribution and introduce random noise to model parameters during training [23]. The Byzantine client adds these perturbations to the updates to mislead the training process and reduce the model performance. Label flip attacks modify the client datasets to carry out targeted attacks on the global models [1]. The attack involves changing the category of each instance in the dataset to the target of misclassified attacks. We compare the attack robustness of FedHyb with a number of methods including the baseline method FedAvg, the client distillation method FedHKD, and two server distillation methods FedDF and DaFKD with a high heterogeneity degree of $\alpha = 0.5$. In all methods, the global accuracy decreases as the number of malicious clients (attackers) increases. FedHyb significantly leads the model robustness under both types of data poisoning attacks.

Real-World Dataset. We use a medical image dataset CheXpert[7] to test the performance of the method in real-world applications. This is a large dataset containing 224316 examples used to interpret chest radiographic images. Table 3 compares the global accuracy of different SOTA methods on CheXpert. It can be seen that the proposed method of FedHyb is able to achieve the highest accuracy, which demonstrates its effectiveness in real-world applications.

4.3 Ablation Study

In Section 3.3, we have evaluated the effects of adding knowledge transfer of logits and features to the server ensemble distillation scheme by visualizing the t-SNE map and the loss landscapes for representation learning and classifier fine-tuning, respectively. Here, we report ablation studies by evaluating the contributions of client and server distillations in FedHyb. Specifically, we study four variants: 1) FedHyb^1 for client self distillation without dynamic weighing, 2) FedHyb^2 for client self distillation with dynamic weighing, 3) FedHyb^3 for server ensemble distillation only without client self distillation, and 4) FedHyb for the proposed two-stage learning paradigm. Table 4 compares the global accuracy results of these variants on CIFAR10 and SVHN with $\alpha = 0.5$. It can be seen that FedHyb^1 performs closely to FedAvg on CIFAR-10 but can be more effective on SVHN on the heterogeneous dataset. With dynamic weighing, FedHyb^2 further improves the local self-distillation process by adjusting it to the client contributions. FedHyb^3 even outperforms FedHyb^2 by including server ensemble distillation. This is consistent with the previous observation [21] that the primary cause of the poor performance on heterogeneous long-tailed data is a biased classifier rather than less generalized features. The complete FedHyb scheme with hybrid knowledge transfer further boosts the server ensemble distillation performance by incorporating better learning abilities of the local client models. This also supports the test results in Table 1 where the two-stage learning method of FedHyb outperforms a simple straightforward combination of two client and server distillation methods by FedHKD+DaFKD.

Table 4. Ablation study on the distillation components of FedHyb.

Method	FedAvg	FedHyb^1	FedHyb^2	FedHyb^3	FedHyb
CIFAR10	0.4741	0.4843	0.4967	0.5425	**0.5531**
SVHN	0.4948	0.6274	0.6351	0.6858	**0.6902**

5 CONCLUSION

In this paper, we proposed a two-stage learning framework called FedHyb for FL on heterogeneous and long-tailed client data. The framework coordinates the seemingly contradicting strategies of instance-balanced sampling and class-balanced sampling by hybrid knowledge distillation. The proposed method can achieve better test accuracy and convergence speed, especially with a higher level of data heterogeneity and an increasing number of distributed clients. The method does not require information exchange between clients and is more robust under poisoning attacks compared to other FL methods designed for heterogeneous data.

Acknowledgements

This work was supported in part by Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515120058).

References

[1] E. Alharbi, L. S. Marcolino, A. Gouglidis, and Q. Ni. Robust federated learning method against data and model poisoning attacks with heterogeneous data distribution. In *Eur. Conf. Artif. Intell.*, pages 85–92, 2023.

[2] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Adv. Neural Inf. Process. Syst.*, pages 1567–1578, 2019.

[3] H. Chen and W. Chao. FedBE: Making Bayesian model ensemble applicable to federated learning. In *Int. Conf. Learn. Represent.*, pages 1–21, 2020.

[4] H. Chen and H. Vikalo. Federated learning in Non-IID settings aided by differentially private synthetic data. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5026–5035, 2023.

[5] H. Chen, Johnny, Wang, and H. Vikalo. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. In *Int. Conf. Learn. Represent.*, pages 1–24, 2023.

[6] Y. J. Cho, A. Manoel, G. Joshi, R. Sim, and D. Dimitriadis. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In *Int. Jt. Conf. Artif. Intell.*, pages 2881–2887, 2022.

[7] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *AAAI Conf. Artif. Intell.*, pages 590–597, 2019.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Int. Conf. Mach. Learn.*, pages 5132–5143, 2020.

[9] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. In *M.Sc. Thesis, Univ. Toronto.*, 2009.

[10] D. Li and J. Wang. FedMD: Heterogenous federated learning via model distillation. In *Adv. Neural Inf. Process. Syst. - Work. Fed. Learn. Data Priv. Confidentiality*, pages 1–8, 2019.

[11] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Adv. Neural Inf. Process. Syst.*, pages 6389–6399, 2018.

[12] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10713–10722, 2021.

[13] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In *Mach. Learn. Syst.*, volume 2, pages 429–450, 2020.

[14] X. Li, W. Yang, Z. Zhang, K. Huang, and S. Wang. On the convergence of FedAvg on Non-IID data. In *Int. Conf. Learn. Represent.*, pages 1–26, 2020.

[15] X. Liao, W. Liu, C. Chen, P. Zhou, H. Zhu, Y. Tan, J. Wang, and Y. Qi. HyperFed: Hyperbolic prototypes exploration with consistent aggregation for Non-IID data in federated learning. In *Int. Jt. Conf. Artif. Intell.*, pages 3957–3965, 2023.

[16] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Adv. Neural Inf. Process. Syst.*, pages 2351–2363, 2020.

[17] P. Liu, X. Yu, and J. T. Zhou. Meta knowledge condensation for federated learning. In *Int. Conf. Learn. Represent.*, pages 1–14, 2023.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Int. Conf. Artif. Intell. Stat.*, pages 1273–1282, 2017.

[19] L. Muñoz-González, K. Co, and E. Lupu. Byzantine-robust federated machine learning through adaptive model averaging. In *arXiv: Machine Learning*, 2019.

[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Adv. Neural Inf. Process. Syst.*, pages 2118 – 2126, 2011.

[21] X. Shang, Y. Lu, G. Huang, and H. Wang. Federated learning on heterogeneous and long-tailed data via classifier re-training with federated features. In *Int. Jt. Conf. Artif. Intell.*, pages 2218–2224, 2022.

[22] J. Shao, F. Wu, and J. Zhang. Selective knowledge sharing for privacy-preserving federated distillation without a good teacher. *Nat. Commun.*, 15(349):1–11, 2024.

[23] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu. Data poisoning attacks against federated learning systems. In *IEEE Eur. Symp. Secur. Priv.*, page 480–501, 2020.

[24] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(86):2579–2605, 2008.

[25] W. Wan, S. Hu, J. Lu, L. Y. Zhang, H. Jin, and Y. He. Shielding federated learning: Robust aggregation with adaptive client selection. In *Int. Jt. Conf. Artif. Intell.*, pages 753–760, 2022.

[26] H. Wang, Y. Li, W. Xu, R. Li, Y. Zhan, and Z. Zeng. DaFKD: Domain-aware federated knowledge distillation. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 20412–20421, 2023.

[27] Q. Xu, T. Cohn, and O. Ohrimenko. Fingerprint attack: Client de-anonymization in federated learning. In *Eur. Conf. Artif. Intell.*, pages 2792–2801, 2023.

[28] T. Yang, S. Zhu, and C. Chen. GradAug: A new regularization method for deep neural networks. In *Adv. Neural Inf. Process. Syst.*, volume 33, pages 14207–14218, 2020.

[29] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Comput. Surv.*, 56(3):1–44, 2024.

[30] H. Zhang and Q. Yao. Decoupling representation and classifier for noisy label learning. In *Int. Conf. Learn. Represent.*, pages 1–16, 2020.

[31] H. Zhang, J. Liu, J. Jia, Y. Zhou, H. Dai, and D. Dou. FedDUAP: Federated learning with dynamic update and adaptive pruning using shared data on the server. In *Int. Jt. Conf. Artif. Intell.*, pages 2776–2782, 2022.

[32] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan. Fine-tuning global model via data-free knowledge distillation for Non-IID federated learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10164–10173, 2022.

[33] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng. Deep long-tailed learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(9): 10795–10816, 2023.

[34] Z. Zhu, J. Hong, and J. Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Int. Conf. Mach. Learn.*, pages 12878–12889, 2021.