DaFKD: Domain-aware Federated Knowledge Distillation

Haozhao Wang[1], Yichen Li[1,3], Wenchao Xu[4], Ruixuan Li[1]*, Yufeng Zhan[5] and Zhigang Zeng[2]
{[1]School of Computer Science and Technology,[2]School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology}, Wuhan, China
[2]Soochow University, Suzhou, China, and [3]The Hong Kong Polytechnic University, Hong Kong
[4]Beijing Institute of Technology, Beijing, China
{hz_wang, rxli, zgzeng}@hust.edu.cn, wenchxu@polyu.edu.hk, and yu-feng.zhan@bit.edu.cn

Abstract

Federated Distillation (FD) has recently attracted increasing attention for its efficiency in aggregating multiple diverse local models trained from statistically heterogeneous data of distributed clients. Existing FD methods generally treat these models equally by merely computing the average of their output soft predictions for some given input distillation sample, which does not take the diversity across all local models into account, thus leading to degraded performance of the aggregated model, especially when some local models learn little knowledge about the sample. In this paper, we propose a new perspective that treats the local data in each client as a specific domain and design a novel domain knowledge aware federated distillation method, dubbed DaFKD, that can discern the importance of each model to the distillation sample, and thus is able to optimize the ensemble of soft predictions from diverse models. Specifically, we employ a domain discriminator for each client, which is trained to identify the correlation factor between the sample and the corresponding domain. Then, to facilitate the training of the domain discriminator while saving communication costs, we propose sharing its partial parameters with the classification model. Extensive experiments on various datasets and settings show that the proposed method can improve the model accuracy by up to 6.02% *compared to state-of-the-art baselines.*

1. Introduction

Federated learning (FL) has emerged as a prominent distributed machine learning framework to train a global model via the collaboration among users without sharing their original dataset [17, 22, 27]. Due to the benefits of preserving privacy and economic communication efficiency, FL has been widely adopted in various applications such as medical image processing [8, 19, 32] and recommendation [2, 26].

The classic FL paradigm, FedAvg [22], iteratively optimize the global model by aggregating the parameters of local models trained from data resides on a number of remote devices or servers. However, these methods usually suffer from serious model performance degradation when the data is not independently and identically distributed (Non-IID) across clients, which is a common issue in FL scenarios. This is mainly because the model parameters on different clients are optimized towards diverse directions [14], leading to the overlarge variance of the aggregated model.

To tackle this challenge, federated distillation (FD) [13] proposes to distill the knowledge of multiple local models into the global model by aggregating only the output soft predictions, which recently attracts increasing attention. For instance, Tao *et al*. [18] leveraged the public dataset as the distillation data samples to obtain the soft predictions from multiple local models and then updated the global model with the average of these soft predictions. Based on [18], Zhu *et al*. [40] and Zhang *et al*. [37] improved the distillation by replacing the public dataset with data generated by the generative model. Although these methods achieve significant improvement over existing parameter-averaging methods, they still do not take the model diversity into account and may still limit the model performance. More specifically, only computing the average of soft predictions will inevitably bring errors when some local models make wrong predictions for the distillation sample.

To break the limitations of existing federated distillation methods, we in this paper propose a novel federated distillation method dubbed DaFKD that can discern the importance of each model to the given distillation sample, and thus is able to reduce the impact of wrong soft predictions. More specifically, we consider that the local data in each client constitutes as a specific domain and employ a domain discriminator for each client to identify the correlation factor between the sample and the domain. For a given distillation sample, we endow the local model with high im-

*Ruixuan Li is the corresponding author.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

portance when its correlation factor is significant and vice versa. The principle behind this is the fact that a model tends to make the correct prediction when the sample is contained in the domain for training the model. Furthermore, to facilitate the training of domain discriminator, we propose sharing its partial parameters with the target classification model. Through extensive experiments over various datasets (MNIST, EMNIST, FASHION MNIST, SVHN) and different settings (various models and data distributions), we show that the proposed method significantly improves the model accuracy as compared to state-of-the-art algorithms. The contributions of this paper are:

- We propose a new domain aware federated distillation method named DaFKD which endows the model with different importance according to the correlation between the distillation sample and the training domain.
- To adaptively discern the importance of multiple local models, we propose employing the domain discriminator for each client which identifies the correlation factors. To facilitate the training of the discriminator, we further propose sharing partial parameters between the discriminator and the target classification model.
- We establish the theories for the generalization bound of the proposed method. Different from existing methods, we theoretically show that DaFKD efficiently solves the Non-IID problem where the generalization bound of DaFKD does not increases with the growth of data heterogeneity.
- We conduct extensive experiments over various datasets and settings. The results demonstrate the effectiveness of the proposed method which improves the model accuracy by up to 6.02% compared to state-of-the-art methods.

2. Related Work

Federated Learning with Parameters Aggregation The main problem incurred by Non-IID for the model parameters aggregation based methods is the huge diverse of the parameters of local models across clients [21,39], where the local models are optimized to different directions. To tackle this challenge, many prior works seek to to reduce the diverse across local model parameters for efficient model aggregation [1, 36]. For instance, Tian *et al*. [16] proposed adding a regularization item in the local objective function such that the divergence of the local model is constrained by the global model. Sai *et al*. [14] proposed reducing the variance of local gradient to align the diverse local update. This paper applies the ensemble distillation over the obtained local models which is orthogonal to these methods.

Knowledge Distillation is to transfer the knowledge from one or more networks (teacher) to another (student) [12]. The key step in knowledge distillation is to align the soft prediction of the student model to that of the teacher model [5, 25, 30, 33, 38]. For example, some works leverage a proxy dataset to distill knowledge between networks [11, 34]. Considering that the proxy dataset may not always exist, some recent works proposed distillation in a data-free manner including reconstructing samples used for training the teacher [20, 23] or learning a generator [35]. By following this path forward, we in this paper particularly focus on the distillation in federated learning by tailoring the generative model and ensemble distillation techniques.

Federated Distillation Federated distillation is to distill the knowledge from multiple teacher models trained by different clients to the student model [3, 9, 28]. Lin *et al*. [18] first proposed leveraging the knowledge distillation in the server to transfer the knowledge from multiple local models to the global model based on an unlabeled proxy dataset. Chen *et al*. [4] proposed linearly aggregating multiple local models with weights generated by the Bayesian posterior to produce a series of combined models and then, distilling these models into one global model. Considering that these methods rely on an unlabeled auxiliary dataset in the server which may not exist in real-world settings, Zhu *et al*. [40] and Zhang *et al*. [37] proposed replacing the proxy dataset with data generated by the generative models and making ensemble federated distillation in a data-free way. However, most of these methods construct the ensemble knowledge by simply computing the average of soft predictions from multiple local models, which does not take model diversity into account and may limit the model performance. In this paper, we take the domain knowledge for training local models into account and endow the local models with different importance when ensembling these soft predictions.

3. Methodology

In this section, we specify the proposed method DaFKD. As illustrated in Figure 1, the key idea of the DaFKD is to leverage a domain discriminator to discern the importance of each local model to the given distillation sample such that the performance of ensemble distillation can be improved. Specifically, each client in DaFKD trains the local model with its private dataset and the domain discriminator with a global generator in an adversarial way. Then, the server aggregates local models from all participated clients and make ensemble distillation with the generator producing distillation samples and the discriminator producing importance. Besides, to facilitate the training of the discriminator, we further propose sharing partial parameters between the discriminator and the target classification model. The workflow of the proposed algorithm is shown in the Algorithm 1.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

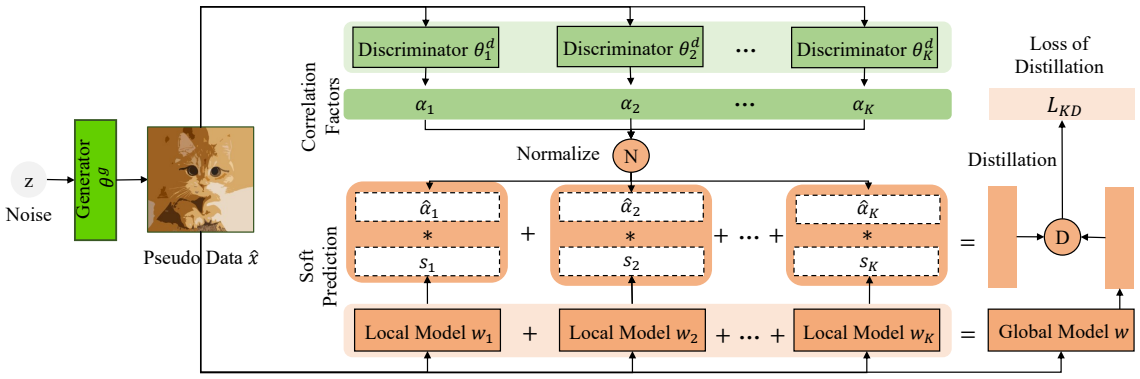


Figure 1. Illustration of the DaFKD framework in the server. The distillation data samples $\hat{x}$ are generated by the generator θ^g. By inputting the pseudo sample into the domain discriminator θ_k^d, the correlation factor α_k of each domain is obtained. Then, the soft predictions s_k are obtained by inputting the data into the local models w_k and are scaled with the correlation factors $\hat{\alpha}$. Finally, the server aggregates all scaled soft predictions and computes distillation loss $\mathcal{L}_{KD}$.

3.1. Problem Formulation

We aim to collaboratively train a global model for K total clients in FL. We consider each client k can only access to his local private dataset $\mathbb{D}_k := \{x_i^n, y_i\}$, where x_i is the i-th input data sample and $y_i \in \{1, 2, \cdots, C\}$ is the corresponding label of x_i with C classes. We denote the number of data samples in dataset $\mathbb{D}_k$ by D_k. The global dataset is considered as the composition of all local datasets $\mathbb{D} = \{\mathbb{D}_1, \mathbb{D}_2, \cdots, \mathbb{D}_K\}$, $D = \sum_{k=1}^{K} D_k$. The objective of the FL learning system is to learn a global model w that minimizes the total empirical loss over the entire dataset $\mathbb{D}$:

$$\min_{w} \mathcal{L}(w) := \sum_{k=1}^{K} \frac{D_k}{D} \mathcal{L}_k(w),$$
$$\text{where } \mathcal{L}_k(w) = \frac{1}{D_k} \sum_{i=1}^{D_k} \mathcal{L}_{CE}(w; x_i, y_i), \tag{1}$$

where $\mathcal{L}_k(w)$ is the local loss in the k-th client and $\mathcal{L}_{CE}$ is the cross-entropy loss function that measures the difference between the prediction and the ground truth labels.

3.2. Domain Discriminator

To endow the model with suitable importance to the given distillation sample, an intuition is that the model has high probability in making the correct prediction when the sample is contained in the domain for training the model. As a consequence, quantifying the correlation between the domain and any given sample is necessary. Motivated by the adversarial training techniques [7] where the discriminator is trained to distinguish whether the generated data is sampled from the distribution of the target dataset, we in this paper proposes a domain discriminator which views the local dataset as the target and outputs the correlation between the distillation sample and the local domain.

More specifically, we assign a personalized discriminator θ_k^d for each client k and adopt a global generator θ^g shared by all clients. At each round t, each participated client k firstly pulls the generator θ^g from the server to produce pseudo dataset $\hat{\mathbb{D}}_k$ with $\hat{D}_k$ samples by sampling noise from the distribution p_z. Then, each client k labels the samples in local private dataset $\mathbb{D}_k$ positive and the samples in pseudo dataset $\hat{\mathbb{D}}_k$ negative. Using these data samples with generated labels, each client k trains the domain discriminator θ_k^d with the following loss function:

$$\min_{\theta_k^d} \mathcal{L}_{adv}^k(\theta_k^d) = \frac{-1}{D_k + \hat{D}_k} \Big[\sum_{x_i \in \mathbb{D}_k} \log f(\theta_k^d; x_i) + \sum_{x_i \in \hat{\mathbb{D}}_k} \log\left(1 - f(\theta_k^d; x_i)\right) \Big], \tag{2}$$

where $f(\theta_k^d; x_i)$ denotes the probability of x_i being real data. Considering that there have been extensive works for training effective generators [37, 40] which can be jointly used with our method, we in this paper simplify the training of generator and adopt the basic FedAvg [22] to train the generator, which still exhibits great effectiveness. Specifically, after obtaining the discriminator θ_k^d, the client k in turn leverages the discriminator to train the generator, which maximizes the loss function 2:

$$\max_{\theta_k^g} \mathcal{L}_{adv}^k(\theta_k^g) = -\mathbb{E}_{z \sim p_z(z)} \log\left(1 - f(\theta_k^d; g(\theta_k^g; z))\right). \tag{3}$$

It is worthwhile to note that the client can also train the generator θ_k^g and the discriminator θ_k^d in an alternative way like [7]. After obtaining the updated local generator θ_k^g, the server receives them from all K_t participated clients and aggregates them to get the new global generator:

$$\theta^g = \frac{1}{K_t} \sum_{k=1}^{K_t} \theta_k^g. \tag{4}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

From a global perspective, the adversarial loss function can be formulated as:

$$\begin{aligned}&\max_{\theta^g}\min_{\theta_1^d,\cdots,\theta_K^d}\mathcal{L}_{adv}(\theta_1^d,\cdots,\theta_K^d)\\&=-\frac{1}{K}\sum_{k=1}^{K}\Big[\mathbb{E}_{x\sim p_k(x)}\log f(\theta_k^d;x)\\&+\mathbb{E}_{z\sim p_z(z)}\log\left(1-f(\theta_k^d;g(\theta^g;z))\right)\Big].\end{aligned}\tag{5}$$

Discussion. One concern for this method may be that the local generator will lead to privacy leakage when it is uploaded from client to the server. In fact, there have been many prior works solving this problem. For example, to protect privacy, Zhu *et al*. [40] proposed generating feature maps instead of the original data and Xin *et al*. [31] proposed exploiting the differential privacy. Similarly, the privacy concern of the global generator can also be addressed by only allowing the generator to output intermediate features as specified in Zhu *et al*. [40]. The main method proposed in this paper supports various generators and thus can definitely achieve privacy protection as combined with these privacy-protecting methods together, of which more details can be found in Appendix A.

3.3. Domain-aware Federated Distillation

To obtain the classification model, in each round t, each participated client k firstly locally trains the model w_t^k and sends it as well as the domain discriminator θ_k^d to the server. After receiving the two models, the server aggregates multiple local models w_t^k by computing their average as:

$$\hat{w}_{t+1}=\frac{1}{K_t}\sum_{k=1}^{K_t}w_t^k.\tag{6}$$

Then, the server uses the global generator θ^g to generate the pseudo dataset $\hat{\mathbb{D}}^g$ as the distillation data. After that, for every distillation sample $x_i\in\hat{\mathbb{D}}^g$, the server computes the importance of each local model w_t^k with the domain discriminator θ_k^g as $\alpha_{k,i}=f(\theta_k^d;x_i)$ and then normalizes it into the probability:

$$\hat{\alpha}_{k,i}=\frac{f(\theta_k^d;x_i)}{\sum_{k=1}^{K_t}f(\theta_k^d;x_i)},\tag{7}$$

which guarantees that $\sum_{k=1}^{K_t}\hat{\alpha}_{k,i}=1$. Finally, the server inputs the pseudo sample x_i into each local model w_t^k and the average model $\hat{w}_{t+1}$ to obtain the soft predictions $s(w_t^k;x_i)$ and $s(\hat{w}_{t+1};x_i)$, and applies ensemble knowledge distillation with the importance $\hat{\alpha}_{k,i}$ to obtain the global model w_{t+1}:

$$\begin{aligned}w_{t+1}&=\arg\min_{\hat{w}_{t+1}}\mathcal{L}_{KD}(\hat{w}_{t+1})\\&=\frac{1}{\hat{D}^g}\sum_{x_i\in\hat{\mathbb{D}}^g}KL\Big(\sum_{k=1}^{K_t}\hat{\alpha}_{k,i}\cdot s(w_t^k;x_i),s(\hat{w}_{t+1};x_i)\Big),\end{aligned}\tag{8}$$

where $KL(\cdot)$ is to compute the Kullback-Leibler divergence (KL-divergence).

Discussion. In our method, the domain discriminator is not necessary to be uploaded to the server. Since the domain discriminator is mainly used to output the importance of each model to the distillation sample, it can definitely be used in each client when the distillation sample is generated locally. Naturally, considering that each client has access to the global generator which is used to train the domain discriminator, the shared distillation dataset can be generated in each local client when the random seed of noise is consistent. Besides, to further protect the privacy of importance, each client can directly upload the soft predictions weighted by importance to the server, which we defer the

Algorithm 1: Workflow of DaFKD Algorithm

Input : the learning rate η
Output: final classification model w_T

In server:
Initialize classification model w_1 and generator θ_1^g;
for $t=1$ ***to*** T **do**
  randomly select K_t clients from K total clients;
  for *each selected client* k **in parallel do**
    sends w_t and θ_t^g to the client k;
    receives w_t^k, θ_k^d, and θ_k^g from the client k;
  end
  aggregates generators θ_k^g with (4) to get θ_{t+1}^g;
  aggregates local models w_t^k with (6) to get $\hat{w}_{t+1}$;
  produces pseudo samples $\hat{\mathbb{D}}^g$ with generator θ_k^g;
  obtain correlation factors $\hat{\alpha}$ with (7);
  distills knowledge with (8) to get w_{t+1};
end
In each selected client k:
receives w_t and θ_t^g from the server;
set local models $w_t^k=w_t$ and $\theta_k^g=\theta_t^g$;
set θ_k^d as the same in previous round;
for $e=1$ **to** E **do**
  randomly draws a mini-batch of samples x_k;
  generates a mini-batch of pseudo samples $\hat{x}_k$;
  updates w_t^k and θ_k^d with (9);
  samples a mini-batch of noise samples z_k;
  updates local generator θ_k^g with (3);
end
pushes the w_t^k, θ_k^d, and θ_k^g to the server;

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

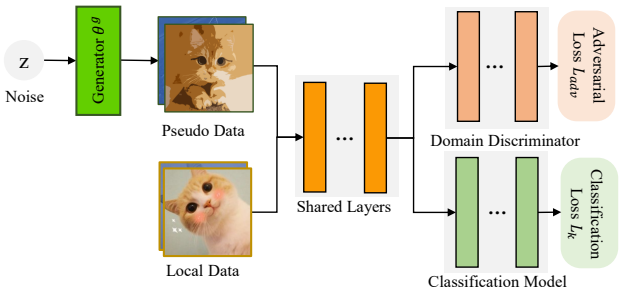


Figure 2. Illustration of the Shared Learning between Discriminator and Classification Model.

details to the Appendix B.

3.4. Partial Parameters Sharing

Considering that the domain discriminator is trained over the local dataset, its performance may be deteriorated when the size of local dataset is small. Motivated by the idea in the multi-task learning where sharing the encoder between different tasks can promote each other, we propose sharing partial parameters between the discriminator θ_k^d and the classification model w^k to solve this problem. The intuition behind this is that both the discriminator and the classification model have to distinguish the sample from the extracted features. Besides, another benefit of sharing layers is that the communication cost can be reduced when the discriminator is also uploaded to the server.

To this end, we propose sharing the front model layers between the two models which are used to extract features, as illustrated in Figure 2. Specifically, by denoting the c-layer shared extractor by $\tilde{w}^k = [\tilde{w}^{k,1}, \ldots, \tilde{w}^{k,c}]$ where $\tilde{w}^{k,i}$ is the i-th layer of the shared extractor, we can re-write the n_d-layer discriminator by $\theta_k^d = [\tilde{w}^k, \theta_k^{d,c+1}, \ldots, \theta_k^{d,n_d}]$, and the n-layer classification model w^k by $w^k = [\tilde{w}^k, w^{k,c+1}, \ldots, w^{k,n}]$ with $w^{k,i}$ as the i-th layer of the classification model $w^{k,i}$. With the shared layers, the discriminator and the classification model are trained in a joint way:

$$\min_{\theta_k^d, w^k} \mathcal{L}_J^k(\theta_k^d, w^k) = \mathcal{L}_{adv}^k(\theta_k^d) + \mathcal{L}_k(w^k), \tag{9}$$

where $\mathcal{L}_{adv}^k(\theta_k^d)$ refers to (2) and $\mathcal{L}_k(w^k)$ is defined in (1).

3.5. Theoretical Analysis

In this section, we prove that our method can solve the Non-IID problem of FD. To verify this, we first analyze the distribution learned by the generator and discriminator.

Theorem 1 *Denote the data distribution of each client k by $p_k(x)$, the data distribution of all clients by $p(x)$, and the pseudo data distribution of the generator by $p_g(x)$. If the Algorithm 1 trains the discriminator θ_k^d and the global generator θ^g to the optima for the loss function (5), then the pseudo data distribution of the generator is $p_g^*(x) = p(x)$, and the discriminator outputs $f^*(\theta_k^d; x) = \frac{p_k(x)}{p_k(x)+p(x)}$ for each client $k = 1 \ldots K$.*

The detailed proof is deferred to Appendix C. Theorem 1 exhibits that the global generator still learns the global data distribution even though there are multiple discriminators specialized for different clients. Besides, the discriminator can distinguish the samples either from the global distribution or the local distribution, and thus can generate efficient correlation factors with the collaboration of the generator that produces global distribution. For simplicity of notation, we denote the local model $f(w_k; x)$ trained over the local dataset $\mathbb{D}_k$ by $h_{\hat{p}_k}(x)$ and $f(w; x)$ trained over global dataset $\mathbb{D}$ by $h_{\hat{p}}(x)$. Besides, without losing generality, we consider $D_1 = \cdots = D_K = m$. Then, we can further derive the generalization bound of the proposed method.

Theorem 2 *Denote the empirical distribution of activation from each client k by $\hat{p}_k$ and the empirical distribution of global dataset by $\hat{p} = \frac{1}{K}\sum_{k=1}^K \hat{p}_k$. Then, given the constants $0 < \delta \leq 1$ and $\sigma > 0$, with the probability at least $1-\delta$, the expected generalization error $\mathcal{L}_p(\sum_{k=1}^K \hat{\alpha}_k(x) h_{\hat{p}_k})$ of domain-aware ensemble model is:*

$$\begin{aligned} &\mathcal{L}_p(\sum_{k=1}^K \hat{\alpha}_k(x) h_{\hat{p}_k}) \\ &\leq (K+1)\mathcal{L}_{\hat{p}}(h_{\hat{p}}) + (K+1)\sqrt{\frac{\sigma^2 log\frac{2K}{\delta}}{2m}}. \end{aligned} \tag{10}$$

The proof can be found in the Appendix D. For ease of comparison, we here present the bounds of state-of-the-art FD methods. Specifically, the bound of FEDFUSION [18] is

$$\begin{aligned} &\mathcal{L}_p(\sum_{k=1}^K \hat{\alpha}_k(x) h_{\hat{p}_k}) \\ &\leq \mathcal{L}_{\hat{p}}(h_{\hat{p}}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}} + \frac{1}{2K}\sum_{i=1}^K d_{\mathcal{H}\Delta\mathcal{H}}(p_k, p) + \lambda_k, \end{aligned} \tag{11}$$

where $\lambda_k = \min_{h\in\mathcal{H}} \mathcal{L}_{p_k}(h) + \mathcal{L}_p(h)$ and FEDGEN [40] is

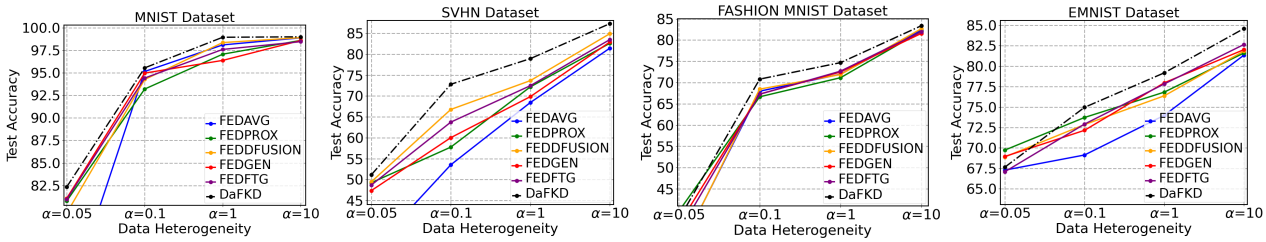
$$\begin{aligned} &\mathcal{L}_p(\sum_{k=1}^K \hat{\alpha}_k(x) h_{\hat{p}_k}) \leq \mathcal{L}_{\hat{p}}(h_{\hat{p}}) \\ &+ \sqrt{\frac{4}{m'}(d\log\frac{2em}{d} + \log\frac{4K}{\delta})} + \frac{1}{K}\sum_{i=1}^K d_{\mathcal{H}\Delta\mathcal{H}}(p'_k, p) + \lambda'_k. \end{aligned} \tag{12}$$

The main differences between DaFKD and baselines are the items $\frac{1}{K}\sum_{i=1}^K d_{\mathcal{H}\Delta\mathcal{H}}(p'_k, p)$ and λ_k which measure the distance between the local data distribution and the global distribution and are incurred by the Non-IID. DaFKD does not include the two items, which indicates that the Non-IID problem is efficiently solved.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

Top-1 Test Accuracy							
Dataset	Setting	FEDAVG	FEDPROX	FEDDFUSION	FEDGEN	FEDFTG	DaFKD
MNIST, E = 20	$\alpha = 0.05$	69.11± 1.39	80.77± 0.35	79.42± 0.57	81.06± 1.09	80.95±1.06	**82.33±0.44**
	$\alpha = 0.1$	95.16± 0.79	93.21± 0.55	94.27± 0.12	94.98± 0.47	94.43±0.49	**95.56±0.41**
	$\alpha = 1$	98.11± 0.14	97.08± 0.69	98.37± 0.40	96.39± 0.90	98.47±0.21	**98.96±0.38**
SVHN, E = 20	$\alpha = 0.05$	33.01± 0.12	49.24± 0.16	49.46± 0.17	47.36± 0.42	48.69±1.87	**51.14±0.16**
	$\alpha = 0.1$	53.54± 0.21	57.77± 0.86	66.78± 0.33	60.03± 1.12	63.75±0.11	**72.80±0.11**
	$\alpha = 10$	81.44± 0.01	82.61± 0.34	84.91± 0.64	82.91± 0.73	83.49±1.32	**87.31±0.85**
FASHION MNIST, E = 20	$\alpha = 0.05$	30.01± 0.54	**39.71±0.23**	30.08± 0.82	36.59± 0.98	34.84±0.77	37.85±0.24
	$\alpha = 0.1$	67.97± 0.03	66.65± 0.08	68.46± 0.14	67.29± 2.05	67.25±0.14	**70.81±0.21**
	$\alpha = 10$	82.37± 0.82	82.06± 0.53	82.67± 1.03	81.57± 1.96	81.96±1.86	**83.37±0.06**
EMNIST, E = 40	$\alpha = 0.05$	67.28± 0.14	69.73± 0.17	68.89± 0.07	**68.95±0.88**	67.08±0.97	67.64±1.86
	$\alpha = 0.1$	69.13± 0.23	73.72± 0.55	72.85± 0.93	72.15± 2.04	72.91±1.87	**74.96±0.91**
	$\alpha = 10$	81.35± 1.03	81.61± 0.71	81.85± 1.08	82.02± 1.19	82.65±1.04	**84.60±1.86**

Table 1. Performance of our DaFKD and other baseline methods on four image datasets. For all methods, a smaller α indicates higher heterogeneity and E indicates the local training steps.



(a) MNIST Dataset. (b) SVHN Dataset. (c) FASHION MNIST Dataset. (d) EMNIST Dataset.

Figure 3. Visualized performance w.r.t data heterogeneity.

4. Experiments

In this section, we compare the performance of our proposed approach with related works.

4.1. Setup

Baselines: In addition to **FEDAVG** [22], **FEDPROX** regularizes the local model training with a proximal term in the model objective [16]. **FEDDFUSION** is a data-based knowledge distillation method, which applies unlabeled training samples as the proxy dataset [18]. **FEDGEN** is a data-free knowledge distillation approach that each client can directly regulate the local model updating using the generated unlabeled samples in server [40]. **FEDFTG** learns a generator to ensemble knowledge of local models in a data-free manner and fine-tunes the global model in server instead of broadcasting the aggregated model back to each client directly [37].

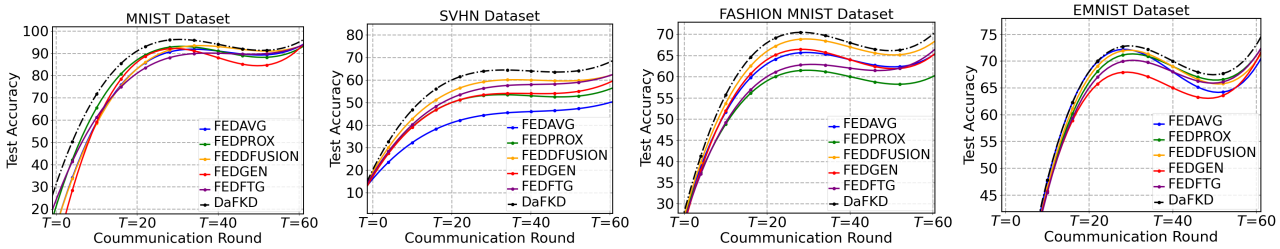
Dataset: We conduct experiments on four image datasets with heterogeneous dataset partition: **MNIST** [15], **EMNIST** [6], **FASHION MNIST** [29] and **SVHN** [24]. Among them, MNIST, EMNIST and SVHN dataset is for digit and character image classifications, and FASHION MNIST is a fashion-product dataset which is used to learn a multi-class classification task.

Configurations: Unless otherwise mentioned, we set the number of local training epoch $E = 20$, communication round $T = 60$, the client number $K = 20$ with an active ratio $r = 0.4$. For local training, the batch size is 32 and the weight decay is $1e-3$. The learning rate is 0.01 for distillation and 0.001 for training the classifier, generator, and discriminator. Like FEDGEN [40], we use the Dirichlet distribution Dir(α) on labels to simulate the data heterogeneity. We apply all the training samples and distribute them to user models, and we use all the testing samples for the performance evaluation. For the classifier in all methods, we employ ResNet11 [10] as the basic backbone. For the generator in FEDGEN, FEDFTG and DaFKD, we apply the network composed of two embedding layers(for one-hot label vector and noise vector respectively) and the fully-connected layer with LeakyReLU and BatchNorm layers. For the multi-task learning structure in our DaFKD approach, we treat all previous layers before the last fully-connected layer as share layers, and we use two different fully-connected layers to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

		Communication Round					
Dataset	Accuracy	FEDAVG	FEDPROX	FEDDFUSION	FEDGEN	FEDFTG	DaFKD
MNIST	$acc = 85\%$	22.67±2.33	**18.33±3.67**	19.67±8.33	21.67±2.00	20.67±1.33	**19.00±2.67**
	$acc = 90\%$	**33.33±1.00**	40.00±3.33	46.33±2.33	39.00±3.67	43.67±3.67	**38.33±1.67**
SVHN	$acc = 55\%$	58.33±6.67	50.67±3.33	**21.67±3.33**	32.67±5.67	30.00±4.67	**14.00±2.33**
	$acc = 60\%$	> 60	> 60	**40.67±2.00**	57.33±3.67	55.67±2.33	**18.67±1.33**
FASHION MNIST	$acc = 60\%$	21.00±1.33	22.67±5.67	**20.67±3.33**	25.00±3.33	27.67±4.67	**18.67±2.33**
	$acc = 65\%$	35.67±3.67	38.33±4.00	**34.33±0.67**	39.67±2.67	43.33±6.66	**33.67±3.00**
EMNIST	$acc = 65\%$	**16.33±6.33**	**18.00±3.33**	21.33±5.67	23.33±1.67	22.67±3.67	20.00±3.33
	$acc = 70\%$	57.66±1.33	44.67±2.67	42.67±4.67	50.67±2.33	**41.33±0.67**	**40.67±4.33**

Table 2. Evaluation of DaFKD and other baseline methods on four image datasets ($\alpha = 0.1$), in terms of the communication rounds to reach the target test accuracy (acc). Here we highlight the **best** and **second-best** results in bold.



(a) MNIST Dataset. (b) SVHN Dataset. (c) FASHION MNIST Dataset. (d) EMNIST Dataset.

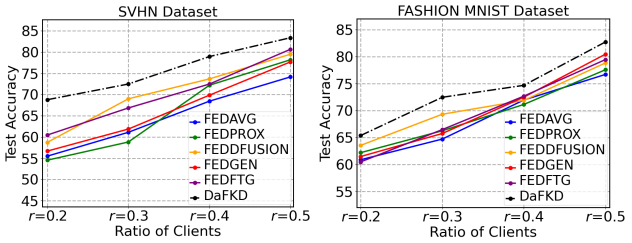
Figure 4. Fitted learning curve of four image datasets in 60 communication rounds ($\alpha = 0.1$).

get outputs as the classifier result and discriminator result.

4.2. Performance Overview

Test accuracy. Table 1 shows the performance of our DaFKD and other baseline methods on four image datasets. We carry out experiments against different levels of data heterogeneity on each dataset, and DaFKD achieves the best performance in most cases. Among all mentioned approaches, FEDDFUSION, FEDGEN, FEDFTG and DaFKD apply the knowledge distillation with extra data to improve the model training. As shown in Table 1, these KD based methods are notably excellent, and outperform FEDAVG and FEDPROX in most scenarios. Besides, DaFKD is the only KD-algorithm that is robust against different datasets while consistently performs well, especially surpassing the second by 2-6% with different data heterogeneity levels on SVHN dataset. These results verify our motivation that the domain discriminator can identify the correlation and thus solve the heterogeneity problem.

Data heterogeneity and Client participant. Figure 3 displays the test accuracy with different levels of data heterogeneity on four image datasets. As vividly shown in this figure, all methods achieve an improvement in test accuracy with the decreasing degree of data heterogeneity. Most notably, DaFKD gains a significant improvement in test accuracy in data heterogeneity $\alpha = 0.1$ and steadily outperforms all methods. Figure 5 provides the test accuracy under different ratios r between active clients and total clients. In this figure, DaFKD performs best with different ratios r and the higher accuracy is achieved as applying more active clients in each communication round.

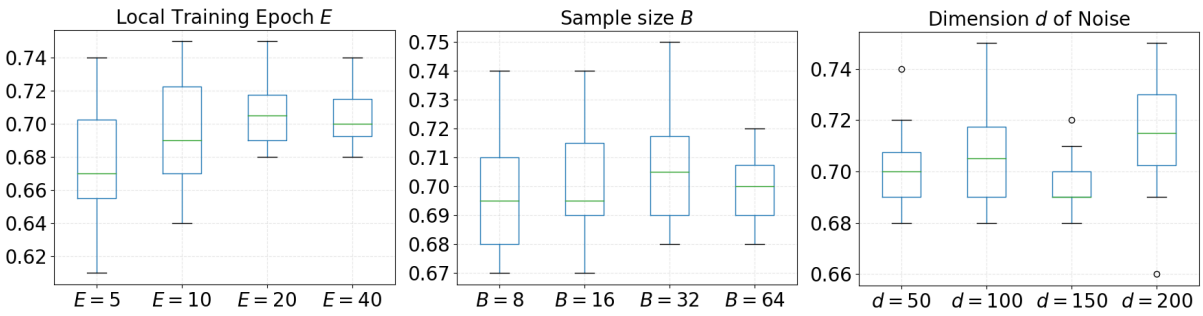


(a) SVHN Dataset. (b) FASHION MNIST Dataset.

Figure 5. Test accuracy w.r.t. ratio r between active clients and total clients in each round ($\alpha = 0.1$).

Communication rounds. Table 2 shows evaluation of DaFKD and other baseline methods in terms of the communication rounds to reach the target test accuracy. Here we highlight the best and second-best results in bold. DaFKD reaches most best and the second-best evaluation results on all datasets. What is more, although FEDAVG reach the target accuracy with fewer communication rounds on EMNIST dataset, DaFKD finally can surpass it by significant 5% after all communication rounds. Meanwhile, Figure 4 displays the fitted learning curve of four image

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1



(a) Box plot w.r.t E. (b) Box plot w.r.t B. (c) Box plot w.r.t d.

Figure 6. Performance of DaFKD under different configurations (a) local training epoch E, (b) sample size B in classifier and generator, (c) dimension d of noise on SVHN with $\alpha = 0.1$.

	$DaFKD$		$DaFKD^{no-sharing}$		$DaFKD^{no-correlation}$	
	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.1$
MNIST	82.33±0.44	**95.56±0.41**	**84.67±0.92**	95.23±0.16	80.57±1.67	94.14± 0.85
SVHN	**51.14±0.16**	72.80±0.11	50.78±0.03	**74.01±1.08**	50.33±2.08	72.51±0.98
FASHION MNIST	**37.85±0.24**	**70.81±0.21**	35.45±0.34	70.51±0.14	34.64±1.68	64.35±1.87
EMNIST	67.64±1.86	**74.96±0.91**	**68.20±0.07**	73.61±2.32	65.01±0.06	71.72±0.67

(a) Test accuracy (%) of DaFKD with different techniques.

		$DaFKD$	$DaFKD^{no-sharing}$	$DaFKD^{no-correlation}$
MNIST	$acc = 85\%$	**19.00±2.67**	21.33±1.67	20.67±1.33
	$acc = 90\%$	**38.33±1.67**	47.00±2.67	45.33±2.67
SVHN	$acc = 55\%$	**14.00±2.33**	22.33±2.67	20.67±1.33
	$acc = 60\%$	**18.67±1.33**	35.33±5.67	31.00±3.67
FASHION MNIST	$acc = 60\%$	**18.67±2.33**	**18.67±1.00**	19.33±0.67
	$acc = 65\%$	33.67±3.00	**33.33±1.67**	37.67±1.67
EMNIST	$acc = 65\%$	**20.00±3.33**	22.33±2.00	21.33±2.33
	$acc = 70\%$	**40.67±4.33**	45.66±1.67	42.33±0.33

(b) The number of communication rounds to reach the given accuracy.

Table 3. Ablation studies.

datasets in 60 communication rounds, where we adapt the test accuracy per five communication rounds to fit the learning curve. In the figure, each method rapidly increases at the beginning and slows down as training goes. DaFKD always keeps a leading tendency in the test accuracy.

Parameter sensitivity analysis. To figure out whether DaFKD is sensitive to some specific parameters, we select local training epoch E, sample batchsize B and dimension d of noise on SVHN with $\alpha = 0.1$ to carry out experiments. Figure 6 shows the performance of DaFKD under different configurations. DaFKD achieves the better result when we increase the local training epochs at the beginning. However, DaFKD has a comparable performance when the E is set to 20 and 40. In order to balance communication expense and test accuracy, we give priority to the local training epoch $E = 20$ here. In addition, DaFKD achieves a similar performance with different sample size B and dimension d of noise. This indicates that DaFKD is only sensitive to few parameters and still robust to most parameters in a large range.

The impact of sharing parameters. We share all front layers (ResNet11) before the last fully-connected layer as share layers between the generator and the discriminator in DaFKD. Table 3 shows the results. As can be seen, parameter sharing facilitates the training of DaFKD in two folds: 1) it enables higher accuracy in most cases; 2) it accelerates convergence where it consistently costs fewer rounds as reaching some given accuracy. Furthermore, parameter sharing can save communication costs by transmitting fewer parameters in each round. Besides, from Table 3.(a), we can find that DaFKD with correlations performs better in all cases (sharing or not). From the last two columns of Table 3.(b), we can find that parameter sharing can accelerate convergence faster than correlations.

5. Conclusion

In this paper, we seek to tackle the data heterogeneity challenge in the federated knowledge distillation. We propose a novel method dubbed domain-aware federated knowledge distillation, namely, DaFKD, which imposes an importance on each local model for some given distillation sample. To quantify the importance, we leverage a domain discriminator to compute the correlation between the distillation sample and the domain for training the local model. Furthermore, to facilitate the training of the domain discriminator, we propose sharing its partial parameters to the classification model. Extensive experiments conducted on various datasets and settings show that our method achieves significant improvement for the model accuracy.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2

[2] Muhammad Ammad-ud-din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *CoRR*, abs/1901.09888, 2019. 1

[3] Ilai Bistritz, Ariana Mann, and Nicholas Bambos. Distributed distillation for on-device learning. In *Proceedings of Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2020. 2

[4] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2

[5] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision, ICCV*, pages 4794–4802, 2019. 2

[6] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 6

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680. 3

[8] Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2423–2432. 1

[9] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 11020–11029, 2020. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[11] Ruifei He, Shuyang Sun, Jihan Yang, Song Bai, and Xiaojuan Qi. Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 9151–9161. 2

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[13] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479, 2018. 1

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, pages 5132–5143. 1, 2

[15] Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 2010. 6

[16] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 2, 6

[17] Xin-Chun Li, Yi-Chu Xu, Shaoming Song, Bingshuai Li, Yinchuan Li, Yunfeng Shao, and De-Chuan Zhan. Federated learning with position-aware neurons. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10072–10081. 1

[18] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020. 1, 2, 5, 6

[19] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1013–1023. 1

[20] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *CoRR*, abs/1710.07535, 2017. 2

[21] Grigory Malinovskiy, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtárik. From local SGD to local fixed-point methods for federated learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML, 13-18 July, Virtual Event*, pages 6692–6701, 2020. 2

[22] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 3, 6

[23] Paul Micaelli and Amos J. Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9547–9557. 2

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[25] Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1364, 2019. 2

[26] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *CoRR*, abs/1906.04329, 2019. 1

[27] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. ATPFL: automatic trajectory prediction model design under federated learning framework. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR,*

New Orleans, LA, USA, pages 6553–6562, June 18-24, 2022. 1

[28] Guile Wu and Shaogang Gong. Peer collaborative learning for online knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, 2021. 2

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[30] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10687–10698, 2020. 2

[31] Bangzhou Xin, Wei Yang, Yangyang Geng, Sheng Chen, Shaowei Wang, and Liusheng Huang. Private FL-GAN: differential privacy synthetic data generation based on federated learning. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 2927–2931. 4

[32] An Xu, Wenqi Li, Pengfei Guo, Dong Yang, Holger Roth, Ali Hatamizadeh, Can Zhao, Daguang Xu, Heng Huang, and Ziyue Xu. Closing the generalization gap of cross-silo federated medical image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 20834–20843. 1

[33] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, volume 33, pages 5628–5635, 2019. 2

[34] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 4633–4642. 2

[35] Jaemin Yoo, Minyong Cho, Taebum Kim, and U Kang. Knowledge extraction with no observable data. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2701–2710. 2

[36] Lin Zhang, Yong Luo, Yan Bai, Bo Du, and Ling-Yu Duan. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *IEEE/CVF International Conference on Computer Vision, ICCV, Montreal, QC, Canada, October 10-17, 2021*, pages 4400–4408, 2021. 2

[37] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10164–10173. 1, 2, 3, 6

[38] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 4320–4328, 2018. 2

[39] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018. 2

[40] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pages 12878–12889. PMLR, 2021. 1, 2, 3, 4, 5, 6

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

A. DaFKD Without Uploading the Discriminator

The privacy of local generators can be protected by using secure aggregation. The privacy of the global generator can be protected by only outputting features instead of the original data, as shown in Figure 7, which is elaborated in FEDGEN[40] and mentioned in Section 3.2.

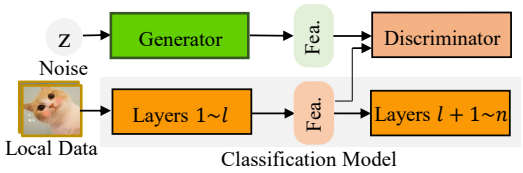


Figure 7. Feature generator. The discriminator and the generator learn the feature map instead of the original dataset. Similarly, the distillation dataset also includes the feature map.

B. DaFKD Without Uploading the Discriminator

In fact, the discriminator and the correlation factors are not necessarily visible to the server to protect the privacy of clients. More specifically, all clients can use the same generator to produce pseudo distillation data locally. Then, each client k inputs the distillation data x_i to the discriminator θ_k^d to produce correlation factors $f(\theta_{k,d}, x_i)$ and input the distillation data to the classification model w^k to produce soft predictions $s_{k,i}$. To enable the domain-aware federated distillation, each client k multiplies the correlation factors $f(\theta_{k,d}, x_i)$ to the corresponding soft predictions s obtaining $f(\theta_{k,d}, x_i)s_{k,i}$ and transmits it to the server. At the same time, the server aggregates $f(\theta_{k,d}, x_i)$ from all clients in a privacy-preserving manner by using differential privacy or homomorphic encryption to obtain $\sum_{k=1}^{K_t} f(\theta_{k,d}, x_i)$. After receiving multiplied soft predictions $\alpha_{k,i}s_{k,i}$ from all clients and the aggregated $\sum_{k=1}^{K_t} f(\theta_{k,d}, x_i)$, the server normalizes the multiplied soft predictions getting $\frac{\alpha_{k,i}s_{k,i}}{\sum_{k=1}^{K_t} f(\theta_{k,d}, x_i)}$. To enable distillation, the server uses the same random seed as each client is adopted to produce the pseudo data x_i and inputs it to the global model $\hat{w}$ obtaining $s(\hat{w}; x_i)$. Finally, the server implements the ensemble distillation using (8), i.e.,

$$w_{t+1} = \arg\min_{\hat{w}_{t+1}} \mathcal{L}_{KD}(\hat{w}_{t+1}) = \frac{1}{\hat{D}^g} \sum_{x_i \in \hat{\mathbb{D}}^g} KL\Big(\sum_{k=1}^{K_t} \hat{\alpha}_{k,i} \cdot s(w_t^k; x_i), s(\hat{w}_{t+1}; x_i)\Big).$$

C. Proof of Theorem 1

Theorem 1 *Denote the data distribution of each client k by $p_k(x)$, the data distribution of all clients by $p(x)$, and the pseudo data distribution of the generator by $p_g(x)$. If the Algorithm 1 trains the discriminator θ_k^d and the global generator θ^g to the optima for the loss function (5), then the pseudo data distribution of the generator is $p_g^*(x) = p(x)$, and the discriminator outputs $f^*(\theta_k^d; x) = \frac{p_k(x)}{p_k(x)+p(x)}$ for each client $k = 1 \dots K$.*

Proof: To analyze the distribution fitted by the global generator and multiple discriminators, we formally present the overall adversarial loss function including the generator and all discriminators as:

$$\max_{\theta^g} \min_{\theta_1^d, \cdots, \theta_K^d} \mathcal{L}_{adv}(\theta_1^d, \cdots, \theta_K^d) = -\frac{1}{K} \sum_{k=1}^{K} \Big[\mathbb{E}_{x \sim p_k(x)} \log f(\theta_k^d; x) + \mathbb{E}_{z \sim p_z(z)} \log\left(1 - f(\theta_k^d; g(\theta^g; z))\right)\Big], \quad (13)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

where $p_k(x)$ is the data distribution of client k. Given the fixed generator θ^g, considering the distribution of generated data as $p_g(x)$, we have

$$\begin{aligned}\min_{\theta_1^d,\cdots,\theta_K^d} \mathcal{L}_{adv}(\theta_1^d,\cdots,\theta_K^d) &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\mathbb{E}_{x\sim p_k(x)}\log f(\theta_k^d;x) + \mathbb{E}_{x\sim p_g(x)}\log\left(1-f(\theta_k^d;x)\right)\Big] \\ &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\int_x p_k(x)\log f(\theta_k^d;x)dx + \int_x p_g(x)\log\left(1-f(\theta_k^d;x)\right)dx\Big] \\ &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\int_x p_k(x)\log f(\theta_k^d;x) + p_g(x)\log\left(1-f(\theta_k^d;x)\right)dx\Big].\end{aligned} \tag{14}$$

Obviously, the equation (14) achieves the minima when

$$f^*(\theta_k^d;x) = \frac{p_k(x)}{p_k(x)+p_g(x)},\ \forall k = 1,\cdots,K. \tag{15}$$

Now, to solve the optimal generator, we bring (15) back to (13) and obtain

$$\begin{aligned}\max_{\theta^g}\mathcal{L}_{adv}(\theta^g) &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\mathbb{E}_{x\sim p_k(x)}\log\frac{p_k(x)}{p_k(x)+p_g(x)} + \mathbb{E}_{x\sim p_g(x)}\log\frac{p_g(x)}{p_k(x)+p_g(x)}\Big] \\ &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\int_x p_k(x)\log\frac{p_k(x)}{p_k(x)+p_g(x)}dx + \int_x p_g(x)\log\frac{p_g(x)}{p_k(x)+p_g(x)}dx\Big] \\ &= -\frac{1}{K}\sum_{k=1}^{K}\Big[\int_x p_k(x)\log\frac{p_k(x)}{p_k(x)+p_g(x)} + p_g(x)\log\frac{p_g(x)}{p_k(x)+p_g(x)}dx\Big] \\ &= -\int_x\frac{1}{K}\sum_{k=1}^{K}\Big[p_k(x)\log\frac{p_k(x)}{p_k(x)+p_g(x)} + p_g(x)\log\frac{p_g(x)}{p_k(x)+p_g(x)}\Big]dx \\ &= \log 4 - \frac{1}{K}\sum_{k=1}^{K}\mathrm{JSD}(p_k(x)||p_g(x)),\end{aligned} \tag{16}$$

where JSD denotes the Jensen-Shannon Divergence. Since the centroid defined as the average sum of a finite set of probability distributions is the minimizer of Jensen-Shannon divergences between a probability distribution and the prescribed set of distributions, we can derive the formulation of optimal $p_g(x)$ as $p_g^*(x) = \frac{1}{K}\sum_{k=1}^{K}p_k(x)$, which completes the proof.

D. Proof of Theorem 2

Theorem 2 *Denote the empirical distribution of activation from each client k by $\hat{p}_k$ and the empirical distribution of global dataset by $\hat{p} = \frac{1}{K}\sum_{k=1}^{K}\hat{p}_k$. Then, given the constants $0 < \delta \leq 1$ and $\sigma > 0$, with the probability at least $1-\delta$, the expected generalization error $\mathcal{L}_p(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k})$ of domain-aware ensemble model is:*

$$\begin{aligned}&\mathcal{L}_p(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k}) \\ &\leq (K+1)\mathcal{L}_{\hat{p}}(h_{\hat{p}}) + (K+1)\sqrt{\frac{\sigma^2 log\frac{2K}{\delta}}{2m}}.\end{aligned} \tag{17}$$

Proof: We seek to establish the relationship between $\mathcal{L}_p(\frac{1}{K}\sum_{k=1}^{K}\hat{\alpha}_k h_{\hat{p}_k})$ and $\mathcal{L}_{\hat{p}}(h_{\hat{p}})$. Considering that the convexity of the loss function in terms of the prediction, we have

$$\mathcal{L}_p(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k}) = \int_x p(x)L(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k}(x))dx \leq \int_x p(x)\big[\sum_{k=1}^{K}\hat{\alpha}_k(x)L(h_{\hat{p}_k}(x))\big]dx. \tag{18}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

Considering the optimal discriminator $f^*(\theta_k^d; x) = \frac{p_k(x)}{p_k(x)+p(x)}$ where $p(x) = \frac{1}{K}\sum_{k=1}^{K} p_k(x)$, we have

$$\begin{aligned}
\hat{\alpha}_k(x) &= \frac{f(\theta_k^d; x)}{\sum_{k=1}^{K} f(\theta_k^d; x)} = \frac{\frac{p_k(x)}{p_k(x)+p(x)}}{\sum_{k=1}^{K} \frac{p_k(x)}{p_k(x)+p(x)}} \\
&= \frac{p_k(x)}{(p_k(x)+p(x))\sum_{i=1}^{K} \frac{p_i(x)}{p_i(x)+p(x)}} \\
&\leq \frac{p_k(x)}{\frac{p_k(x)+p(x)}{\max\{p_1(x),\cdots,p_K(x)\}+p(x)} \sum_{i=1}^{K} p_i(x)} \\
&\leq \frac{p_k(x)}{\frac{p(x)}{Kp(x)+p(x)} \sum_{i=1}^{K} p_i(x)} \\
&= (K+1)\frac{p_k(x)}{\sum_{i=1}^{K} p_i(x)} \\
&= \frac{(K+1)}{K} \cdot \frac{p_k(x)}{p(x)}.
\end{aligned} \tag{19}$$

Bringing the bound of $\hat{\alpha}$ in (19) back to (18) derives:

$$\begin{aligned}
\mathcal{L}_p(\sum_{k=1}^{K} \hat{\alpha}_k(x) h_{\hat{p}_k}) &\leq \int_x p(x)\big[\sum_{k=1}^{K} \frac{(K+1)}{K} \cdot \frac{p_k(x)}{p(x)} L(h_{\hat{p}_k}(x))\big] dx \\
&= \frac{(K+1)}{K} \sum_{k=1}^{K} \int_x p_k(x) L(h_{\hat{p}_k}(x)) dx \\
&= \frac{(K+1)}{K} \sum_{k=1}^{K} \mathcal{L}_{p_k}(h_{\hat{p}_k}).
\end{aligned} \tag{20}$$

Next, we bound the $\mathcal{L}_{p_k}(h_{\hat{p}_k})$ with its empirical counterpart $\mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k})$ through Hoeffding inequality. Without losing the generality, we consider the simplified case where the size of samples in all clients are equal, i.e., $D_1 = D_2 = \ldots = D_K = m$. Then, a simple application of the Hoeffding's inequality gives:

$$P(|\mathcal{L}_{p_k}(h_{\hat{p}_k}) - \mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k})| \geq \epsilon) \leq 2\text{exp}(-\frac{2m\epsilon^2}{\sigma^2}), \tag{21}$$

where $\epsilon > 0$ and $\sigma > 0$ are the constants. Thereby, with probability at least $1 - \frac{\delta}{K}$, we have:

$$\mathcal{L}_{p_k}(h_{\hat{p}_k}) \leq \mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}. \tag{22}$$

For all K devices, we have

$$\begin{aligned}
&P\Big[\bigcap_{k=1}^{K} \big(\mathcal{L}_{p_k}(h_{\hat{p}_k}) \leq \mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}\big)\Big] \\
&= 1 - P\Big[\bigcup_{k=1}^{K} \big(\mathcal{L}_{p_k}(h_{\hat{p}_k}) \geq \mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}\big)\Big] \\
&\geq 1 - \sum_{k=1}^{K} P\Big[\big(\mathcal{L}_{p_k}(h_{\hat{p}_k}) \geq \mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}\big)\Big] \\
&\geq 1 - \delta.
\end{aligned} \tag{23}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 March 2023 doi:10.20944/preprints202303.0432.v1

Putting (22) back to (20) derives:

$$\mathcal{L}_p(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k}) \leq \frac{(K+1)}{K}\sum_{k=1}^{K}\Big(\mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + \sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}\Big). \tag{24}$$

Considering that $h_{\hat{p}_k}$ minimizes the loss function over the distribution $\hat{p}_k$ of training dataset $\mathbb{D}_k$, $\mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) \leq \mathcal{L}_{\hat{p}_k}(h_{\hat{p}})$ can be easily obtained. According to the definition that $\hat{p} = \frac{1}{K}\sum_{k=1}^{K}\hat{p}_k$, we can derive

$$\frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) \leq \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\hat{p}_k}(h_{\hat{p}}) = \mathcal{L}_{\hat{p}}(h_{\hat{p}}). \tag{25}$$

Thereby, the following inequality holds with probability at least $1-\delta$:

$$\begin{aligned}\mathcal{L}_p(\sum_{k=1}^{K}\hat{\alpha}_k(x)h_{\hat{p}_k}) &\leq \frac{(K+1)}{K}\sum_{k=1}^{K}\mathcal{L}_{\hat{p}_k}(h_{\hat{p}_k}) + (K+1)\sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}} \\ &\leq (K+1)\mathcal{L}_{\hat{p}}(h_{\hat{p}}) + (K+1)\sqrt{\frac{\sigma^2 \log\frac{2K}{\delta}}{2m}}.\end{aligned} \tag{26}$$