FedFTG, 2022

Fine-tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning

Lin Zhang[1,4*] Li Shen[2] Liang Ding[3] Dacheng Tao[2,3] Ling-Yu Duan[1,4†]
[1] Peking University, Beijing, China [2] JD Explore Academy, Beijing, China
[3] The University of Sydney, Sydney, Australia [4] Peng Cheng Laboratory, Shenzhen, China
{zhanglin.imre, lingyu}@pku.edu.cn, {mathshenli, dacheng.tao}@gmail.com
ldin3097@sydney.edu.au

Abstract

Federated Learning (FL) is an emerging distributed learning paradigm under privacy constraint. Data heterogeneity is one of the main challenges in FL, which results in slow convergence and degraded performance. Most existing approaches only tackle the heterogeneity challenge by restricting the local model update in client, ignoring the performance drop caused by direct global model aggregation. Instead, we propose a data-free knowledge distillation method to fine-tune the global model in the server (FedFTG), which relieves the issue of direct model aggregation. Concretely, FedFTG explores the input space of local models through a generator, and uses it to transfer the knowledge from local models to the global model. Besides, we propose a hard sample mining scheme to achieve effective knowledge distillation throughout the training. In addition, we develop customized label sampling and class-level ensemble to derive maximum utilization of knowledge, which implicitly mitigates the distribution discrepancy across clients. Extensive experiments show that our FedFTG significantly outperforms the state-of-the-art (SOTA) FL algorithms and can serve as a strong plugin for enhancing FedAvg, FedProx, FedDyn, and SCAFFOLD.

1. Introduction

With the explosive growth of data and the strict privacy-protection policy, reckless data transmission and aggregation gradually become unacceptable due to the high bandwidth cost and risk of privacy leakage. Recently, Federated Learning (FL) [30, 31] has been proposed to replace the traditional heavily centralized learning paradigm and protect data privacy. It has been successfully applied in real-world tasks, such as smart city [16, 34, 42], health care [8, 26, 27], and recommender system [9, 10], etc.

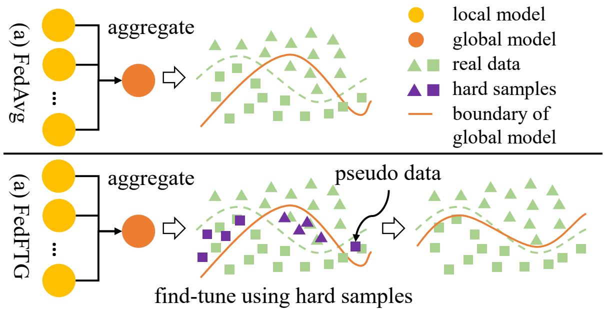


Figure 1. Comparison between FedAvg [30] and FedFTG. By fine-tuning the global model using generated hard samples, FedFTG alleviates the performance decrease after model aggregation.

One of the main challenges in FL is the data heterogeneity, i.e., the data in clients are non-identically and independently distributed (Non-IID). It has been verified that the vanilla FL algorithm, FedAvg [30], leads to drifted local models and forgets the global knowledge catastrophically in this scenario, which further induces degraded performance and slow convergence [14, 19, 21]. This is because the local model is updated merely with local data, i.e., minimizing the local empirical loss. However, minimizing the local empirical loss is fundamentally inconsistent with minimizing the global empirical loss [1, 24, 29] in Non-IID FL.

To tackle the data heterogeneity challenge, most existing methods, e.g., FedProx [23], SCAFFOLD [18], FedDyn [1], MOON [22] constrain the direction of local model update to align the local and global optimization objectives. Recently, FedGen [43] learns a lightweight generator to generate pseudo feature and broadcasts it to clients to regulate local training. However, all these methods merely conduct simple model aggregation to get the global model in server, which ignores local knowledge incompatibility and induces knowledge forgetting in the global model. In addition, [36]

[*] Part of this work was done during Ling Zhang's internship at JD Explore Academy.

[†] Corresponding author.

shows that directly aggregating models will largely degrade the performance while fine-tuning can greatly boost the accuracy. These motivate us to fine-tune the aggregated global model in the server with the knowledge in local models. On the other hand, merely aggregating local models in server ignores the server's rich computing resources that could be potentially utilized to improve the performance of FL, such as the computing source in cross-silo FL [17].

Motivated by these observations, we propose a novel approach that boosts the performance of standard FL by on-the-fly fine-tuning the global model via data-free knowledge distillation (FedFTG), which simultaneously refines the model aggregation procedure and exploits the rich computing power of the sever. Concretely, FedFTG models the input space of local models through an auxiliary generator in the server, then generates pseudo data to transfer the knowledge in local models to the global model to improve the performance. To facilitate effective knowledge distillation throughout the training, FedFTG iteratively explores the hard samples in data distribution, which will induce prediction disagreement between local models and global model. Figure 1 compares FedFTG with FedAvg. FedFTG fine-tunes the global model with the hard samples to correct the model shift after model aggregation. The generator and global model are adversarially trained in a data-free manner, thus the whole procedure will not violate the privacy policy in FL. Considering the label distribution shift in data heterogeneity scenario, we further propose customized label sampling and class-level ensemble techniques, which explore the distribution correlation of clients and exploit maximum utilization of knowledge.

FedFTG is orthogonal to several existing local optimizers, such as FedAvg, FedProx, FedDyn, SCAFFOLD and MOON, as it only modifies the procedure of global model aggregation in the server. Consequently, FedFTG can be seamlessly embedded into these local FL optimizers, taking their advantages to further improve the performance of FedFTG. Extensive experiments on various settings verify that FedFTG achieves superior performance compared with state-of-the-art (SOTA) methods.

The main contributions of this work are four-fold:

- We propose FedFTG to fine-tune the global model in server via data-free distillation, which simultaneously enhances the model aggregation step and utilizes the computing power of the server.
- We develop hard sample mining to effectively transfer knowledge to global model. Besides, we propose customized label sampling and class-level ensemble to facilitate maximum utilization of knowledge.
- We demonstrate that FedFTG is orthogonal to exiting local optimizers and can serve as a strong and versatile plugin to enhance the performance of FedAvg, FedProx, FedDyn, SCAFFOLD and MOON.
- We verify the superiority of FedFTG against several SOTA methods for FL, including FedAvg, FedProx, FedDyn, SCAFFOLD, MOON, FedGen and FedDF, with extensive experiments on five benchmarks.

2. Related Work

There exist extensive works on improving the global performance of FL via client selection [4, 7, 15], split learning [11, 38], domain adaptation [26, 33], etc. The readers may refer to monographs [17, 37] and the reference therein to follow up its recent advances. Below, we mainly summarize the most relevant techniques to our work.

Federated Optimizer. The vanilla FL algorithm, i.e. FedAvg [30] periodically aggregates the local models in server and updates the local model with its individual data. FedProx [23] adds a proximal term to the local subproblem to restrict the local update closer to the initial (global) model. SCAFFOLD [18] uses a variance reduction technique to correct the drifted local update. FedDyn [1] modifies the objective of client with linear and quadratic penalty terms to align global and local objectives. In summary, all these methods focus on aligning the local and global model to narrow the distribution drift during the local training without enhancing the global model directly as in FedFTG.

Knowledge Distillation in Federated Learning. With the help of an unlabeled dataset, FedDF [25] proposes an ensemble distillation for model fusion, trains the global model using the averaged logits from local models. FedAUX [35] finds a model initialization for the local models, and weights the logits from local models using (ε, δ)-differentially private certainty scoring. FedBE [3] generates a series of global models from Bayesian perspective using the local models, then summarizes these models into one global model by ensemble knowledge distillation. All these methods rely on an unlabeled auxiliary dataset in the server, while it is unclear to which extent should the auxiliary dataset be related to training data to guarantee effective knowledge distillation. Though FedDF maintains the auxiliary dataset can be replaced with a pretrained generator, it does not instantiate how to acquire the generator.

Data-Free Knowledge Distillation (DFKD). DFKD methods [2, 6] generate pseudo data from a pretrained teacher model, and use them to transfer knowledge of teacher model to another student model. The data is generated by maximizing the response of fake data on teacher model. DeepImpression [32] models the output space of teacher model and recovers the real data by fitting the output space. DeepInversion [40] further optimizes the pseudo data by regularizing the distribution of intermediate feature maps. DAFL [2] and DFAD [6] use a generator to generate data efficiently, where DAFL optimizes the generator by maximizing the response on prediction and feature level, and DFAD uses an adversarial training scheme to exploit

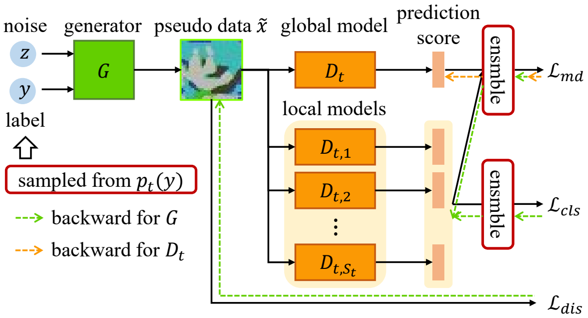


Figure 2. Training procedure of FedFTG in the server. After receiving local models in t-th round and aggregating them, FedFTG adversarially generates hard sample and transfers knowledge to the aggregated global model through $\mathcal{L}_{md}$. $\mathcal{L}_{cls}$ and $\mathcal{L}_{dis}$ are utilized to boost the fidelity and diversity of hard sample. Besides, FedFTG uses customized label sampling and class-level ensemble to derive maximum utilization of knowledge.

the knowledge in teacher model effectively.

FedGen also [43] learns a lightweight generator to ensemble knowledge of local models in a data-free manner, but uses the generator to regularize the local training. Besides, we design hard samples mining scheme, customized label sampling, and class-level ensemble to effectively transfer the knowledge from local models to global model in data heterogeneity scenario.

3. Methodology

In this section, we describe the proposed novel federated learning method: FedFTG. In each communication round, FedFTG randomly selects a set of clients and broadcasts the global model to them. Each client initializes the local model using the global model and trains it with a local optimizer. The server collects the local models and aggregates them as a preliminary global model. Instead of broadcasting the aggregated model back to each client directly, FedFTG fine-tunes this preliminary global model in server using the knowledge extracted from local models. Concretely, we develop a data-free knowledge distillation method with hard sample mining to effectively explore and transfer the knowledge to global model. Considering the label distribution shift in clients, we propose customized label sampling and class-level ensemble to facilitate more effective knowledge utilization. Figure 2 visualizes the training procedure on the server, and the corresponding algorithm is summarized in Algorithms 1&2. Note that FedFTG is orthogonal to efforts on optimizing local model training, such as SCAFFOLD, FedAvg, FedProx, and FedDyn.

3.1. Data-Free Knowledge Distillation With Hard Sample Mining for Global Model Fine-Tuning

Let ω be the model parameter in the server and clients. In this work, we consider there exist K clients, where

Algorithm 1 FedFTG

```
Input: $T$: communication round; $K$: client number; $C$: the fraction of active client in each round; $\{\mathcal{D}_k\}_{k\in\{1,...,K\}}$: the datasets of clients; $\omega$: the parameter of the classifier; $\theta$: the parameter of the generator.
initialize model parameters $\omega$ and $\theta$
for $t = 1, ..., T$ do
    $S_t \leftarrow$ (random set of $\lceil C \cdot K \rceil$ clients);
    for $k \in S_t$ in parallel do
        $\omega_k \leftarrow$ ClientUpdate($\omega, D_k$)
                    ▷ FedAvg, FedProx, FedDyn, and SCAFFOLD
    end for
    $\omega, \theta \leftarrow$ ServerUpdate($\omega, \theta, \{\omega_k\}_{k\in S_t}$)
end for
```

1, 训练local model，做FedAvg

Algorithm 2 ServerUpdate, round t

```
Input: $I$: iteration of the training procedure in server; $I_g$, $I_d$: inner iteration of training the generator and the global model; $\eta_g$: the global step-size; $\omega$, $\theta$, $\{\omega_k\}_{k\in S_t}$, $\lambda_{cls}$, $\lambda_r$.
$\Delta\omega = \frac{1}{|S_t|}\sum_{k\in S_t}(\omega_k - \omega)$, $\omega \leftarrow \omega + \eta_g\Delta\omega$
compute $p_t(y)$ according to Eq. (9)
for $i = 1, ..., I$ do
    $(Z, Y) \leftarrow$ (sample a batch of $z \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and $y \sim p_t(y)$)
    compute $\{\alpha_t^{k,y}\}_{k\in S_t, y\in Y}$ according to Eq. (10)
    for $j = 1, ..., I_g$ do
        update the generator $\theta$ according to Eq. (8) to explore hard samples based on current global model $\omega$
    end for
    for $j = 1, ..., I_d$ do
        update the global model $\omega$ according to Eq. (8) to transfer the knowledge from $\{\omega_k\}_{k\in S_t}$ to $\omega$
    end for
end for
return $\omega, \theta$
```

2, 生成数据

2, 训练Generator model，用于生成hard sample

3，基于生成数据KD generator

3, 根据生成的hard sample fine-tuning(KD) Global Model

3，基于生成数据KD Global

$\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{N_k}$ is the dataset individually stored in k-th client, N_k is the corresponding number of samples. Generally speaking, federated learning can be formulated as the following problem:

$$\min_{\omega} \frac{1}{K}\sum_{k=1}^{K} f_k(\omega),\ f_k(\omega) = \frac{1}{N_k}\sum_{i=1}^{N_k}\mathcal{L}(x_k^i, y_k^i; \omega), \qquad (1)$$

where $\mathcal{L}$ is the loss function to measure training error, and dataset $\mathcal{D}_k$ for each $k \in \{1, 2, ..., K\}$ could be distributed heterogeneously. Due to the privacy protection constraint in FL, the server can not directly access local data of clients. To solve Eq. (1), for each communication round t, existing methods send the global model ω to a random set of clients S_t and optimize it by $\min_\omega f_k(\omega), k \in S_t$. The server collects the local models $\{\omega_k\}_{k\in S_t}$ and aggregates them by averaging the gradients to update the global model ω.

However, the local models are greatly drifted from each other in data heterogeneity scenario. Thus, traditional gra-

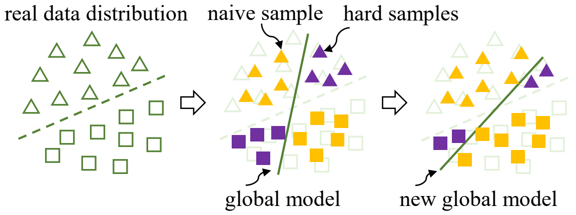


Figure 3. Visualization of hard sample mining. By exploring the hard samples in data distribution and fine-tuning the global model, the global model can be gradually corrected during training.

dient averaging could lose the knowledge in local models, and the performance of updated global model is much lower than local models [44]. To address this issue, we propose a data-free knowledge distillation method to fine-tune the global model, so that the global model can preserve the knowledge in local models and maintain their performance as much as possible. Concretely, the server maintains a conditional generator G that generates pseudo data to capture the data distribution of clients as follows,

生成数据

$$\tilde{x} = G(z, y; \theta), \tag{2}$$

where θ is the parameter of G, $z \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ is a standard Gaussian noise, and y is the class label of $\tilde{x}$ sampled from predefined distribution $p_t(y)$.

As shown in Figure 2, we then input the pseudo data $\tilde{x}$ to the global model to solve the following problem,

$$\min_{\omega} \mathbb{E}_{\substack{z\sim\mathcal{N}(\mathbf{0},\mathbf{1}) \\ y\sim p_t(y)}} [\mathcal{L}_{md}] = \min_{\omega} \mathbb{E}_{\substack{z\sim\mathcal{N}(\mathbf{0},\mathbf{1}) \\ y\sim p_t(y)}} \left[\sum_{k\in S_t} \alpha_t^{k,y} \mathcal{L}_{md}^k \right] \tag{3}$$

where $\mathcal{L}_{md}^k$ is the model discrepancy between global model ω and local model ω_k,

Global Model和Local Model的discrepancy定义为

$$\mathcal{L}_{md}^k = D_{KL}(\sigma(D(\tilde{x}; \omega)) || \sigma(D(\tilde{x}; \omega_k))), \tag{4}$$

where D is the classifier. σ is the softmax function, which will output the prediction score of $\tilde{x}$. D_{KL} denotes the Kullback-Leibler divergence. $\alpha_t^{k,y}$ controls the weight of knowledge from different local models during ensemble. By minimizing $\mathcal{L}_{md}$, we transfer the knowledge in local models to the global model. In Section 3.2 we will introduce how to acquire $p_t(y)$ and $\alpha_t^{k,y}$ to adapt label distribution shift in data heterogeneity scenario.

Data Fidelity and Diversity Constraints. To better extract knowledge from local models, the pseudo data $\tilde{x}$ should fit the input space of local models. Therefore, we use semantic loss $\mathcal{L}_{cls}$ to train the generator G, which will facilitate the fidelity of pseudo data,

$$\min_{\theta} \mathbb{E}_{\substack{z\sim\mathcal{N}(\mathbf{0},\mathbf{1}) \\ y\sim p_t(y)}} [\mathcal{L}_{cls}] = \min_{\theta} \mathbb{E}_{\substack{z\sim\mathcal{N}(\mathbf{0},\mathbf{1}) \\ y\sim p_t(y)}} \left[\sum_{k\in S_t} \alpha_t^{k,y} \mathcal{L}_{cls}^k \right] \tag{5}$$

where $\mathcal{L}_{cls}^k$ is the cross-entropy loss between the prediction of local model on pseudo data $\tilde{x}$ and the class label y,

$$\mathcal{L}_{cls}^k = \mathcal{L}_{CE}(\sigma(D(\tilde{x}; \omega_k)), y), \tag{6}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss. By minimizing $\mathcal{L}_{cls}$, $\tilde{x}$ is enforced to yield higher prediction on class y, thus it fits the data distribution of class y.

Simply using $\mathcal{L}_{cls}$ will lead to model collapse of the generator: G outputs the same data for every class. To address this issue, we use diversity loss $\mathcal{L}_{dis}$ in [43] to improve the diversity of the generated data,

$$\mathcal{L}_{dis} = e^{\frac{1}{Q*Q}\sum_{i,j\in\{1,\ldots,Q\}}\left(-\|\tilde{x}_i - \tilde{x}_j\|_2 * \|z_i - z_j\|_2\right)} \tag{7}$$

where $\tilde{x}_i$ is generated using z_i. By minimizing $\mathcal{L}_{dis}$, the pseudo data will be diverse and scattered in the data space.

Hard Sample Mining. Training the generator G using $\mathcal{L}_{cls}$ will generate pseudo data $\tilde{x}$ with low classification error, which means $\tilde{x}$ contains the most discriminative feature of class y and is easy to be classified. However, these naive samples will not cause the prediction disagreement between global model and local models, i.e., $\mathcal{L}_{md} = 0$, thus the global model is not optimized during training. As illustrated in Figure 3, the naive samples are already correctly classified by the global model. To effectively exploit the knowledge in local models and transfer them to the global model, we explore the hard samples in data distribution that cause prediction disagreement between local models and global model. Concretely, we adversarially train the generator and the global model with $\mathcal{L}_{md}$: (1) the generator is enforced to generate hard samples that maximize $\mathcal{L}_{md}$, and (2) the global model is trained to minimize $\mathcal{L}_{md}$ using the hard samples. As a result, the global model can be gradually fine-tuned to fit the data distribution as in Figure 3.

To the end, the overall objective of FedFTG in the server is formulated as an adversarial learning scheme,

$$\min_{\omega}\max_{\theta} \mathbb{E}_{z\sim\mathcal{N}(\mathbf{0},\mathbf{1}), y\sim p_t(y)} \left[\mathcal{L}_{md} - \lambda_{cls}\mathcal{L}_{cls} - \lambda_{dis}\mathcal{L}_{dis}\right]. \tag{8}$$

先训练Generator生成hard sample，然后再用这些数据训练global model

3.2. Adaptation to Label Distribution Shift for Effective Knowledge Distillation

In data heterogeneity scenario, label distributions are different among clients, i.e., $p^i(y) \neq p^j(y)$ for different clients i and j. This indicates that: (1) the local dataset $\mathcal{D}_k$ of client is class-imbalanced, and the local model trained by $\mathcal{D}_k$ contains imbalanced data information; (2) for one class, the importance of knowledge are different among local models of clients. To facilitate more effective knowledge distillation, we propose customized label sampling and class-level ensemble to adapt these two problems respectively.

Customized Label Sampling. Typically, dataset in local client is class-imbalanced in data heterogeneity scenario, even having no data for some classes. It has been proved that deep neural networks tend to learn the majority classes and ignore the minority classes [5]. Hence, the data information of minority classes in local models could be wrong and misleading, and the generated pseudo data are invalid to measure the model discrepancy. If uniformly sample the class label y, these invalid data will influence the global model training and induce performance decrease. To mitigate this issue, we customize the sampling probability $p_t(y)$ according to the distribution of whole training data in each round, so that more pseudo data with effective information can be generated,

$$p_t(y) \propto \sum_{k\in S_t}\sum_{i=1}^{N_k} \mathbb{E}_{(x_k^i,y_k^i)\sim\mathcal{D}_k}\left[1_{y_i=y}\right] = \sum_{k\in S_t} n_k^y, \tag{9}$$

where $1_{\text{condition}}$ is 1 if the condition is true and 0 otherwise, n_k^y is the instance number of class y in client k. According to Eq. (9), the pseudo data of majority classes have high probability to be generated, thus FedFTG can guarantee effective knowledge distillation in data heterogeneity scenario.

Class-Level Ensemble. The widely used ensemble method in knowledge distillation assigns the same weight to the knowledge from different teacher models [25, 43], i.e., $\alpha_t^{k,y} = \frac{1}{|S_t|}$ in Eq. (3) and Eq. (5). Due to the label distribution shift, for one class the importance of knowledge are different among local models. If assigning the same weight to clients, the important knowledge can not be figured out and utilized properly. Therefore, we propose class-level ensemble, which assigns the ensemble weight according to the individual data distribution of clients. Specifically, we compute the weight $\alpha_t^{k,y}$ via the data proportion of class y in client k against the total data in S_t,

$$\alpha_t^{k,y} = n_k^y / \sum_{i\in S_t} n_i^y, \tag{10}$$

As a result, the knowledge from clients can be flexibly integrated according to their importance on classes, so that FedFTG can facilitate maximum utilization of knowledge from local models.

4. Experiments

In this section, we empirically verify the effectiveness of FedFTG. We summarize the implementation details in Section 4.1, and compare FedFTG with several SOTA FL algorithms in Section 4.2. Ablation studies are conducted to verify the necessity of each component of FedFTG in Section 4.3. To further validate the effect of FedFTG on real-world FL applications, we evaluate the performance of FedFTG on three real-world datasets in Section 4.4.

4.1. Implementation Details

Baselines. We compare FedFTG against FedAvg [30], FedProx [23], SCAFFOLD [18], FedDyn [1], MOON [22], FedGen [43] and FedDF [25]. Since FedDF does not explain how to obtain the generator, we train it in the same way as FedGen.

Datasets. CIFAR10 and CIFAR100 datasets [20] with heterogeneous dataset partition are used to test the efficacy of FedFTG, which are two difficult tasks in FL scenario and are widely adopted in FL research. Similar to existing works [1, 12, 41], we use Dirichlet distribution $\mathbf{Dir}(\beta)$ on label radios to simulate the non-iid data distribution among clients, where a smaller β indicates higher data heterogeneity. During the implementation, we set $\beta = 0.3$ and $\beta = 0.6$.

Network Architecture. For both CIFAR10 and CIFAR100, we employ ResNet18 [13] as the basic backbone. We borrow the generator network architecture from DFAD [6] for FedFTG and FedDF. For FedGen, the network of generator is composed of two embedding layers (for inputs z and y, respectively) and two fully-connected (FC) layers with LeakyReLU and BatchNorm layers between them.

Hyperparameters. For all methods, we set the number of local training epoch $E = 5$, communication round $T = 1000$, the client number $K = 100$ with the active fraction $C = 0.1$ (i.e., $|S_t| = 10$). For local training, the batchsize is 50 and the weight decay is $1e-3$. The learning rates for classifier and generator are initialized to be 0.1 and 0.01 respectively, and they are decayed quadratically with weight 0.998. The dimension of z is 100 for CIFAR10 and 256 for CIFAR100. I, I_g, I_d in Algorithm 2 are 10, 1 and 5, respectively. If not specifically declared, we adopt $\lambda_{cls} = 1.0$ and $\lambda_{dis} = 1.0$, and adopt SCAFFOLD as the FL optimizer in FedFTG.

We further provide detailed implementations and extra experiment results in the supplementary material.

4.2. Performance Comparison

Test Accuracy. Table 1 reports the test accuracy of all compared algorithms on CIFAR10 and CIFAR100 datasets. We provide the performance of centralized learning in the first line. All experiments are repeated over 3 random seeds. In Table 1, FedFTG achieves the best performance in all scenarios, surpassing the second one (i.e., SCAFFOLD) by at least 1.5%. FedDF also employs data-free knowledge distillation to improve the global model in server. It outperforms FedAvg and FedProx, and outperforms FedDyn and MOON in some cases, which further validates the superiority of the scheme "fine-tuning the global model in server". However, it is worse than SCAFFOLD and FedFTG. FedGen yields lower accuracy compared with FedDF and FedFTG, and shows marginal performance gains than FedAvg in some cases. The performance of FedDF and FedGen further verifies the effectiveness of the proposed modules in FedFTG.

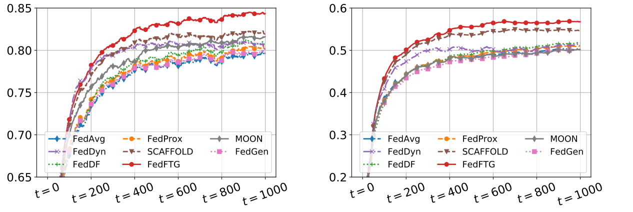
Communication Rounds. Table 2 evaluates different FL methods in term of the number of communication rounds to reach target test accuracy (75% and 80% for CIFAR10, 40% and 50% for CIFAR100, respectively). In Ta-

Table 1. Test Accuracy (%) of different FL methods on CIFAR10 and CIFAR100.

	CIFAR10			CIFAR100		
	iid	$\beta = 0.6$	$\beta = 0.3$	iid	$\beta = 0.6$	$\beta = 0.3$
centralized learning	92.55±0.05			73.98±0.26		
FedAvg	83.78±0.13	82.04±0.46	79.59±1.01	50.29±0.34	50.67±0.34	50.17±0.19
FedProx	84.10±0.39	82.36±0.38	80.12±0.43	51.25±0.62	50.94±0.40	50.82±0.20
FedDyn	85.19±0.58	82.87±0.62	80.15±1.00	53.27±0.01	51.68±0.31	50.51±0.34
MOON	84.34±0.09	82.67±0.08	80.97±0.46	52.51±0.70	52.55±0.49	51.88±0.25
SCAFFOLD	85.99±0.06	84.55±0.30	82.14±1.20	53.32±0.32	53.91±0.33	54.36±0.32
FedGen	83.91±0.36	82.23±0.73	79.72±0.85	50.38±0.27	50.71±0.55	50.08±0.24
FedDF	84.47±0.20	82.92±0.64	80.97±0.74	52.12±0.15	51.36±0.02	51.26±0.09
FedFTG	**87.34**±0.16	**86.06**±0.19	**84.38**±0.49	**56.94**±0.19	**56.49**±0.55	**55.96**±0.39

Table 2. Evaluation of different FL methods on CIFAR10 and CIFAR100 ($\beta = 0.3$), in terms of the number of communication rounds to reach target test accuracy (acc). Note that we highlight the **best** and ***second best*** results in bold.

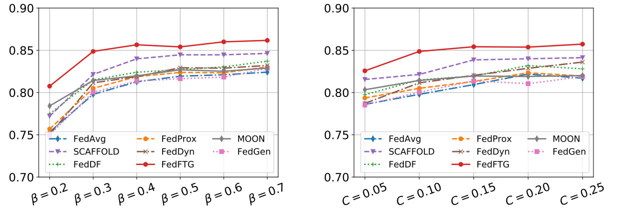
	CIFAR10		CIFAR100	
	$acc = 75\%$	$acc = 80\%$	$acc = 40\%$	$acc = 50\%$
FedAvg	153.67±20.33	425.33±61.67	86.67±6.33	713.67±191.33
FedProx	143.67±0.33	391.67±13.33	86.00±1.00	529.00±36.00
FedDyn	**90.67**±2.33	**183.67**±23.33	64.00±8.00	239.33±15.67
MOON	128.00±10.00	347.00±24.00	79.67±2.33	376.00±29.00
SCAFFOLD	100.33±14.67	212.00±24.00	***58.33***±3.67	***185.67***±0.33
FedGen	140.00±4.00	406.67±29.33	95.00±1.00	684.00±92.00
FedDF	132.67±11.33	329.00±42.00	94.50±1.50	452.00±5.00
FedFTG	***92.67***±14.33	***188.67***±31.33	**57.00**±1.00	**166.33**±10.67



(a) CIFAR10 (b) CIFAR100

Figure 4. Learning Curve of (a) CIFAR10 and (b) CIFAR100 in 1000 communication rounds ($\beta = 0.3$).

ble 2, FedFTG achieves the second best and the best results on CIFAR10 and CIFAR100, respectively. Besides, FedFTG reduces the round number required by its FL optimizer (SCAFFOLD) in all scenarios. For CIFAR10, although FedDyn uses fewer rounds to achieve the target accuracy, its final accuracy is much worse than FedFTG, as displayed in Table 1. Below, we provide the results of using FedDyn as the optimizer of FedFTG, and the derived method FedDyn+FedFTG requires fewer rounds to reach target accuracy than FedDyn. Figure 4 displays the learning curve of different methods in 1000 communication rounds, where FedFTG achieves distinct performance gain after 1000 rounds. Though FedDyn has a faster increase rate in the beginning, the increase trend is gradually slowing down as training goes, and its accuracy is falling behind FedFTG after 150 and 50 rounds for CIFAR10 and CIFAR100 respectively.

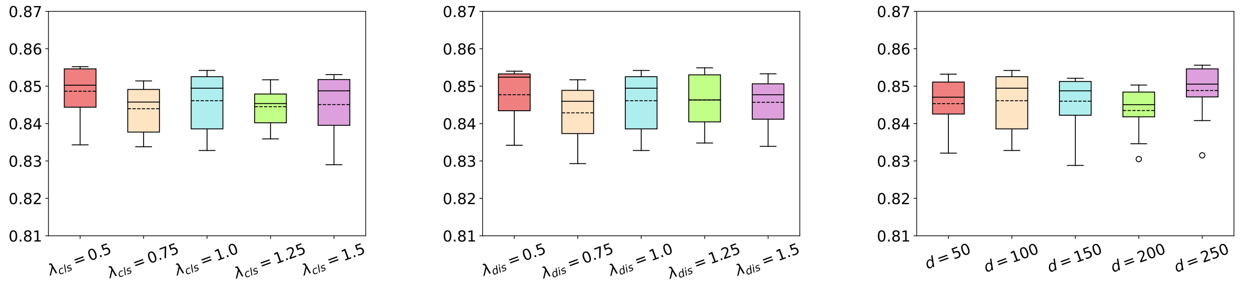


(a) Test accuracy w.r.t. β (b) Test accuracy w.r.t. C

Figure 5. (a) Test accuracy w.r.t. data heterogeneity. (b) Test accuracy w.r.t. fraction C of active clients in each round ($\beta = 0.3$). All experiments are conducted on CIFAR10.

Data heterogeneity and Partial Client Participant. Figure 5(a) displays the test accuracy on different β values. In this figure, FedFTG achieves the best accuracy on all settings, which validates that FedFTG is effective in various data heterogeneity scenarios. Besides, FedFTG gains more accuracy improvement in extreme data heterogeneity scenario $\beta = 0.2$. In addition, as the degree of data heterogeneity decreases i.e., β increases, the accuracy of each method is ascending. Figure 5(b) displays the test accuracy of FL methods with different fractions of active clients in each communication round. FedFTG also yields the best performance in this figure. Besides, the more clients involved in communication, the higher accuracy will be achieved.

Orthogonality of FedFTG with existing FL optimizers. Table 3 provides the performance of FedFTG using FedAvg, FedProx, FedDyn, SCAFFOLD and MOON opti-



(a) Box plot w.r.t. λ_{cls} (b) Box plot w.r.t. λ_{dis} (b) Box plot w.r.t. d

Figure 6. Performance of FedFTG using different hyperparameters (a) λ_{cls}, (b) λ_{dis}, (c) dimension d of noise z on CIFAR10 with $\beta = 0.3$.

Table 3. The impact of FL optimizer on FedFTG.

(a) Test accuracy (%) on CIFAR10, $\beta = 0.3$ and 0.6.

	Accuracy (%)	
	$\beta = 0.6$	$\beta = 0.3$
FedAvg+FedFTG	83.82±0.31	82.27±0.67
FedProx+FedFTG	84.06±0.32	82.21±0.46
FedDyn+FedFTG	83.17±0.50	81.43±0.18
MOON+FedFTG	83.81±0.45	82.19±0.91
SCAFFOLD+FedFTG	**86.06**±0.19	**84.38**±0.49

(b) the round number to reach the target accuracy ($acc = 75\%$ and $acc = 80\%$) when $\beta = 0.3$.

	Round	
	$acc = 75\%$	$acc = 80\%$
FedAvg+FedFTG	122.00±4.00	279.33±17.67
FedProx+FedFTG	117.33±6.67	278.67±25.33
FedDyn+FedFTG	**79.00**±3.00	**168.00**±13.00
MOON+FedFTG	114.33±6.67	276.00±12.00
SCAFFOLD+FedFTG	92.67±14.33	188.67±31.33

Table 4. Impact of the each components in FedFTG. The experiments are conducted on CIFAR10, $\beta = 0.3$.

Method		Accuracy (%)
baseline	FedFTG	83.43±0.10
module	`-hsm`	82.49±0.42
	`-cls`	82.39±0.22
	`-abe`	82.40±0.21
	`-hsm&cls`	82.18±0.16
	`-hsm&abe`	82.15±0.14
	`-cls&abe`	82.11±0.28
	`-hsm&cls&abe`	81.98±0.14
loss	- $\mathcal{L}_{cls}$	82.50±0.55
	- $\mathcal{L}_{dis}$	82.52±0.35
	- $\mathcal{L}_{cls}$- $\mathcal{L}_{dis}$	82.12±0.16
	$D_{KL} \leftarrow \mathcal{L}_{mse}$	10.17±0.26

mizers. In Table 3, SCAFFOLD+FedFTG yields the best test accuracy among all the optimizers. FedDyn+FedFTG performs better than SCAFFOLD+FedFTG in terms of the round number to reach the target accuracy. This is consistent with the results in Table 2, where FedDyn requires fewer rounds than SCAFFOLD. Comparing Table 3 with Tables 1 and 2, we notice that for any FL optimizer, its performance can be largely boosted by using FedFTG. This validates the effectiveness and the orthogonality of FedFTG. Besides, simply using FedAVG+FedFTG as local optimizer already exceeds the other methods in Table 1 except SCAFFOLD.

4.3. Ablation Study

Necessity of each component in FedFTG. Table 4 displays the test accuracy of FedFTG after discarding some modules and losses, trained with 500 communication rounds on CIFAR10, $\beta = 0.3$. Here `hsm`, `cls` and `abe` represent the hard sampling mining, customized label sampling and class-level ensemble, respectively. We can see that removing any module leads to worse and unstable performance, i.e., lower accuracy and larger confidence interval. In addition, their joint absence can cause a further decrease on accuracy. On the other hand, a similar tendency is observed for the losses: the absence of single loss will lead to performance decrease, and removing multiple losses will enlarge the decrease. It should be noticed that, if replacing the KL divergence with Mean Average Square ($\mathcal{L}_{mse}$) to measure the model discrepancy, the model will collapse, which leads to severe performance degradation.

Robustness of FedFTG on hyperparameters. To measure the influence of hyperparameter selection, we select λ_{cls} and λ_{dis} from $[0.5, 0.75, 1.0, 1.25, 1.5]$ and select the dimension d of noise data z in $[50, 100, 150, 200, 250]$. Figure 6 illustrates the test accuracy in term of the box plot, where FedFTG achieves similar performance among all the choices. Besides, the worst accuracy in Figure 6(a)-(b) is better than the best of previous works in Table 1. This indicates that FedFTG is not sensitive to the selection of hyperparameter in a large range.

Comments on feature-level pseudo data. FedGen advises that the data in feature space are more compact than in input space, thus it generates pseudo data in feature-level and fine-tunes the last few FC layers of federated model. Motivated by this, we compare the performance of FedFTG

Table 5. The impact of feature-level generation (F) and input-level generation (I) on FedFTG. The experiments are conducted on CIFAR10, $\beta = 0.3$ and 0.6.

	Accuracy (%)	
	$\beta = 0.6$	$\beta = 0.3$
FedFTG(F)	84.67±0.35	82.76±0.81
FedFTG(I)	86.06±0.19	84.38±0.49

Table 6. Test accuracy (%) on real-world datasets MIP-TCD, Compcar and Tiny-ImageNet.

Method	MIO-TCD	CompCar	Tiny-ImageNet
FedAvg	89.63±1.06	43.34±2.93	34.68±0.67
FedProx	89.69±1.00	44.07±3.41	35.39±0.54
FedDyn	90.47±0.99	50.46±2.57	41.77±0.28
SCAFFOLD	89.88±1.11	48.64±3.46	38.80±0.18
FedGen	89.85±1.03	45.96±4.18	35.44±0.35
FedDF	90.01±0.70	47.31±3.47	36.19±0.40
FedFTG	**91.16±0.92**	**51.85±3.46**	**42.23±0.22**

using feature-level generation (F) and input-level generation (I) in Table 5. Though FedFTG(F) still exceeds the other methods in Table 1, it suffers significant performance drop compared with FedFTG(I), which indicates the input-level generation is more effective for FedFTG. This is because FedFTG(F) only fine-tunes the last few layers of the global model, so the effect of knowledge transfer is limited.

4.4. Experiments on Real-World Datasets

In this section, we test the performance of FedFTG on more challenging real-world datasets - vehicle classification datasets MIO-TCD [28] and CompCar [39], and large-scale image classification dataset Tiny-ImageNet[1]. To better validate the effectiveness of FedFTG, we use the surveillance subset of CompCar, of which the images are collected by surveillance cameras. For MIO-TCD and Tiny-ImageNet, we assign the training data to 100 clients, while for CompCar the client number is 50. β of Dirichlet distribution is 0.6 for all these datasets. The images of MIO-TCD and CompCar are resized to $112 * 112$ before training, and we adopt a deeper generator for them. The communication round is 50, 100 and 1000 for MIO-TCD, CompCar and Tiny-ImageNet respectively. The other settings are the same as in Section 4.1. Experiment results are presented in Table 6.

From this table, we find that FedFTG consistently outperforms the other methods in all scenarios, which verifies the effectiveness of FedFTG in real-world FL applications. FedDF and FedGen also adopt data-free knowledge generation to improve the federated model. Though they yield higher performance than FedAvg and FedProx, FedFTG exceeds them by $1\% \sim 6\%$. This further validates the effectiveness of the proposed modules in FedFTG.

5. Discussion

Privacy issue. Since FedFTG recovers the training data of clients in server, it may violate the privacy regulation in FL. However, according to our observation, the pseudo data only captures the high-level feature pattern of real data, which cannot be understood by human beings (see Figure 2). Besides, as the generator is trained by all local models, the pseudo data tend to show shared features of data in clients, which means the attribute of individual data will not be revealed. Uploading label statistics of data in clients may also leak privacy. One optional solution is adding noise to label statistics. According to our experiments, when the noise ratio is less than 10%, its influence on performance is less than 0.1% on CIFAR10, $\beta = 0.3$ setting.

Communication cost. Compared with other methods, FedFTG only need to additionally transmit the label statistics of training data (i.e., $\{n_t^{k,y}\}_{k \in S_t, y \in [1,..,M]}$, M is the class number), which induces negligibly extra transmission cost. If the training data keep the same during training, the label statistics can be reported to server before training, thus no extra transmission cost will be introduced.

Limitations. The main limitation of this work mainly exists in computation efficiency. As FedFTG additionally trains the global model apart from local training, it will make the whole training time longer than the other methods. In our experiment, FedFTG requires about double the time of FedAVG in each communication round. Besides, as the global model training is conducted in the server, FedFTG is more applicable to the cross-silo FL applications as defined in [17], where the server can be organizations that own sufficient computation source.

6. Conclusion

In this paper we propose a new data-free knowledge distillation method FedFTG to fine-tune the global model and to boost the performance of federated learning. A hard sample mining scheme is proposed to effectively explore the knowledge in local models and transfer it to global model. Facing the label distribution shift in data heterogeneity scenario, we propose customized label sampling and class-level ensemble to derive maximum utilization of knowledge. Extensive experiments on five benchmarks validate the effectiveness of the proposed FedFTG.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62088102, and in part by the PKU-NTU Joint Research Institute (JRI) sponsored by a donation from the Ng Teng Fong Charitable Foundation, and in part by Science and Technology Innovation 2030 –"Brain Science and Brain-like Research" Major Project (No. 2021ZD0201402 and No. 2021ZD0201405).

[1] https://www.kaggle.com/c/tiny-imagenet

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2020. 1, 2, 5

[2] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019. 2

[3] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020. 2

[4] Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020. 2

[5] Cong Fang, Hangfeng He, Qi Long, and Weijie J Su. Layer-peeled model: Toward understanding well-trained deep neural networks. *arXiv preprint arXiv:2101.12699*, 2021. 4

[6] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*, 2019. 2, 5

[7] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. *arXiv preprint arXiv:2105.05883*, 2021. 2

[8] Dashan Gao, Ce Ju, Xiguang Wei, Yang Liu, Tianjian Chen, and Qiang Yang. Hhhfl: Hierarchical heterogeneous horizontal federated learning for electroencephalography. *arXiv preprint arXiv:1909.05784*, 2019. 1

[9] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 1

[10] Florian Hartmann, Sunah Suh, Arkadiusz Komarzewski, Tim D Smith, and Ilana Segall. Federated learning for ranking browser history suggestions. *arXiv preprint arXiv:1911.11807*, 2019. 1

[11] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *arXiv preprint arXiv:2007.14513*, 2020. 2

[12] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020. 5

[13] 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. 5

[14] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 1

[15] Tiansheng Huang, Weiwei Lin, Li Shen, Keqin Li, and Albert Y Zomaya. Stochastic client selection for federated learning with volatile clients. *arXiv preprint arXiv:2011.08756*, 2020. 2

[16] Ji Chu Jiang, Burak Kantarci, Sema Oktug, and Tolga Soyata. Federated learning in smart city sensing: Challenges and opportunities. *Sensors*, 20(21):6230, 2020. 1

[17] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019. 2, 8

[18] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. 1, 2, 5

[19] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 4519–4529. PMLR, 2020. 1

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[21] Gihun Lee, Yongjin Shin, Minchan Jeong, and Se-Young Yun. Preservation of the global knowledge by not-true self knowledge distillation in federated learning. *arXiv preprint arXiv:2106.03097*, 2021. 1

[22] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021. 1, 5

[23] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018. 1, 2, 5

[24] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019. 1

[25] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242*, 2020. 2, 5

[26] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021. 1, 2

[27] Songtao Lu, Yawen Zhang, Yunlong Wang, and Christina Mack. Learn electronic health records by fully decentralized federated learning. *arXiv preprint arXiv:1912.01792*, 2019. 1

[28] Zhiming Luo, Frederic Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new

benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018. 8

[29] Grigory Malinovskiy, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtarik. From local sgd to local fixed-point methods for federated learning. In *International Conference on Machine Learning*, pages 6692–6701. PMLR, 2020. 1

[30] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. pages 1273–1282. PMLR, 2017. 1, 2, 5

[31] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019. 1

[32] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019. 2

[33] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733*, 2019. 2

[34] Basheer Qolomany, Kashif Ahmad, Ala Al-Fuqaha, and Junaid Qadir. Particle swarm optimized federated learning for industrial iot and smart city services. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020. 1

[35] Felix Sattler, Tim Korjakow, Roman Rischke, and Wojciech Samek. Fedaux: Leveraging unlabeled auxiliary data in federated learning. *arXiv preprint arXiv:2102.02514*, 2021. 2

[36] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[37] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021. 2

[38] Guile Wu and Shaogang Gong. Decentralised learning from independent multi-domain labels for person re-identification. *arXiv preprint arXiv:2006.04150*, 2020. 2

[39] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3973–3981, 2015. 8

[40] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 2

[41] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019. 5

[42] Zhaohua Zheng, Yize Zhou, Yilong Sun, Zhang Wang, Boyi Liu, and Keqiu Li. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. *Connection Science*, pages 1–28, 2021. 1

[43] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12878–12889. PMLR, 2021. 1, 3, 4, 5

[44] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 955–963, 2020. 4