Check for
updates

MrTF: model refinery for transductive federated learning

Xin-Chun Li[1,2] · **Yang Yang**[3,4] · **De-Chuan Zhan**[1,2]

Abstract

We consider a real-world scenario in which a newly-established pilot project needs to make inferences for newly-collected data with the help of other parties under privacy protection policies. Current federated learning (FL) paradigms are devoted to solving the data heterogeneity problem without considering the to-be-inferred data. We propose a novel learning paradigm named transductive federated learning to simultaneously consider the structural information of the to-be-inferred data. On the one hand, the server could use the pre-available test samples to refine the aggregated models for robust model fusion, which tackles the data heterogeneity problem in FL. On the other hand, the refinery process incorporates test samples into training and could generate better predictions in a transductive manner. We propose several techniques including stabilized teachers, rectified distillation, and clustered label refinery to facilitate the model refinery process. Abundant experimental studies verify the superiorities of the proposed Model refinery framework for Transductive Federated learning. The source code is available at https://github.com/lxcnju/MrTF.

Keywords Federated learning · Transductive learning · Model refinery

✉ Yang Yang
   yyang@njust.edu.cn

   Xin-Chun Li
   lixc@lamda.nju.edu.cn

   De-Chuan Zhan
   zhandc@nju.edu.cn

[1]   State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, Jiangsu, China

[2]   School of Artificial Intelligence, Nanjing University, Nanjing 210023, Jiangsu, China

[3]   School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China

[4]   Department of Computing, The Hong Kong Polytechnic University, Hong Kong 100872, China

1 Introduction

Machine learning techniques, especially deep learning, have been widely applied to various real-world applications (He et al. 2016; Vaswani et al. 2017; Li et al. 2022; Yang et al. 2021), etc. Inductive learning and transductive learning are two common learning paradigms, where the latter could obtain test samples in advance (Chen et al. 2002; Rohrbach et al. 2013; Liu et al. 2019). Fusing to-be-inferred unlabeled data into training could lead to appreciable performances because it simultaneously captures structural information in both train and test data. The settings of these two paradigms are illustrated at the bottom of Fig. 1.

A practical scenario is that a particular party/user is urgent to make predictions for newly-collected unlabeled data while it has no labeled samples for training. Thus, it needs to seek the help of other relevant parties/users (clients) to build a prediction model in collaboration. There is a restriction that the data of other parties cannot be used directly due to privacy protection policies. Federated Learning (FL) (Yang et al. 2019; McMahan et al. 2017; Li et al. 2022) has been proposed as an efficient distributed training paradigm to collaborate with isolated parties without sending users' data out. On the one hand, various challenges have emerged in FL, e.g., the Non-Independent Identically Distributed (Non-I.I.D.) data challenge (Hsieh et al. 2020). Participating clients may own various data distributions under different contexts, leading to weight divergence in distributed local training (Zhao et al. 2018). On the other hand, existing FL follows an inductive manner that aims to build a global model that could generalize well to any possible forthcoming samples without considering the information of the pre-available test samples. The scenario and challenges are illustrated at the right of Fig. 1, where the newly-established party has 10 classes to infer, while the participating parties may contain only several classes, various classes, or imbalanced classes, etc. These challenges lead to Non-I.I.D. data that hinders the effectiveness of FL.
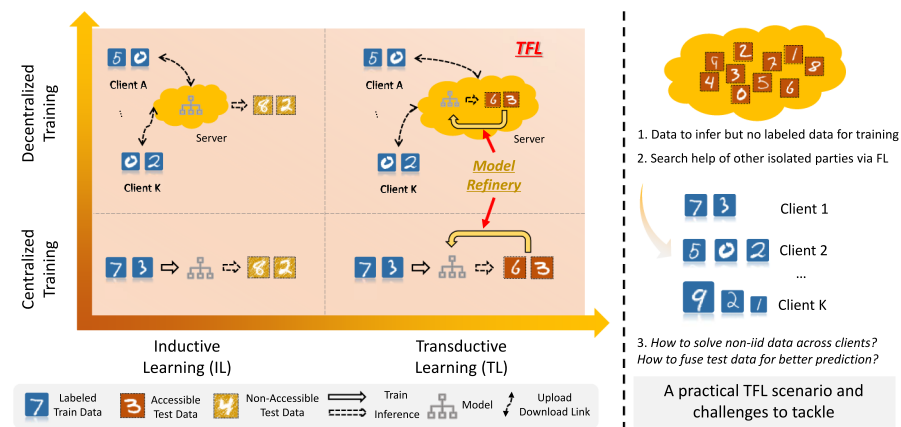

Fig. 1 Left: comparisons of several paradigms. TFL could access test data in advance and the training data is decentralized across clients. Right: a real-world scenario formulated as TFL and corresponding challenges

In this paper, we abstract the scene mentioned above as *transductive federated learning (TFL), where the server owns to-be-inferred data in advance while training data are distributed across clients in a Non-I.I.D. manner*. In TFL, the goal of the server is to assign labels to the test samples on hand. There are two fundamental challenges to tackle: (1) *how to overcome the Non-I.I.D. challenge across clients during distributed training?* (2) *how to improve the inference process for unlabeled data on the server?* To solve the former, some previous FL works take various techniques. For example, some works (Jeong et al. 2018) allow sending out a small portion of clients' data. A better way to meet the privacy policy is the recently proposed FedDF (Lin et al. 2020), which views the local updated models as "teachers" and distills the knowledge to the aggregated model on a publicly available dataset for robust model fusion. However, collecting appropriate public data is also challenging, especially in data-scarce scenarios. For the latter, we should consider the structural information contained in the pre-available test data to assign better predictions. Considering both, FedDF seems to be an appropriate solution. First, the trouble of collecting unlabeled public data is omitted in TFL because the to-be-referred data is a good candidate. What's more, distilling on the to-be-referred samples also considers their information, which is expected beneficial for making predictions.

Nevertheless, it is not all smooth sailing. FedDF encounters several fatal challenges faced with large amounts of clients and stochastic client selection. Specifically, FedDF takes the "AvgLogi" (i.e., Averaging Logits) of local updated models as the ensemble, while the logits' magnitudes across local models vary a lot, and directly averaging them as the "teacher" may lead to training instability. Additionally, stochastic client selection does not guarantee that the local model's ensemble covers all classes, negatively transferring knowledge for missing classes. Correspondingly, we propose a more stable way as an alternative to refine the aggregated model via rectifying the local models' logits and introducing label clustering techniques. Verified on several benchmarks, our proposed methods show superiorities towards other methods. Our contributions could be briefed as originally introducing the practical TFL framework and proposing an effective solution named MrTF.

2 Related works

Our work is closely related to federated learning (FL), transductive learning (TL), and external data in FL.

2.1 Federated learning (FL)

FL (Yang et al. 2019; McMahan et al. 2017) aims to organize isolated clients to accomplish the machine learning process following a distributed training style. As the most standard FL algorithm, Federated Averaging (FedAvg) (McMahan et al. 2017) follows the parameter server architecture (Li et al. 2013) where a server coordinates amounts of clients. During the whole process, only model parameters are transmitted, and advanced privacy protection methods (e.g., differential privacy

(Abadi et al. 2016)) could be additionally applied for stricter privacy protections. The Non-I.I.D. challenge in FL refers to that data of participating users are heterogeneous under various contexts, which hinders the aggregation and personalization in FL (Zhao et al. 2018; Li and Zhan 2021). Various solutions are proposed for better model fusion via introducing local regularization (Li et al. 2020; Yao et al. 2018), reducing gradient variance (Karimireddy et al. 2020), fine-tuning aggregated model via additional data (Jeong et al. 2018; Lin et al. 2020), etc. Knowledge distillation (Hinton et al. 2015) could facilitate the generalization in FL (Afonin and Karimireddy 2022; Zhang et al. 2022). However, these FL methods are only devoted to solving the data heterogeneity problem and could not be directly applied to simultaneously solve the two major challenges of TFL.

2.2 Transductive learning (TL)

Inductive learning assumes test data are not available during training, requiring the trained models to generalize well on any possible test set. As the opposition, TL could access the to-be-inferred data, and the training process could progressively capture the structural information in both train and test data. TL relaxes the requirement of model generalization and only aims to make better predictions on the available test data. Hence, compared with inductive learning, TL could basically achieve better results when the test samples are accessible. TSVM (Chen et al. 2002) utilizes the margin information in test samples and yields better SVM models. A real-world scenario is that we have to make predictions for unlabeled samples in a novel domain, and fusing labeled samples from source domains for together training is a common solution in transfer learning (Rohrbach et al. 2013) or domain adaptation (Long et al. 2015). Another advantageous scene for TL is learning with few-shot samples, where some studies (Liu et al. 2019) have verified the superiorities of TL.

2.3 External data in FL

To reduce the weight divergence in FL, (Jeong et al. 2018) utilize additional labeled data on the server to fine-tune the global model, while (Li and Wang 2019) resorts to publicly available labeled data. Some semi-supervised FL also introduces unlabeled data (Jeong et al. 2021; Long et al. 2021), while they consider the clients own both labeled and unlabeled samples. The most related work to ours is FedDF (Lin et al. 2020), which utilizes "AvgLogi" to ensemble local models and further distill the knowledge from them to the aggregated model. However, FedDF only considers the cross-silo scenes defined in Kairouz et al. (2019) where the amount of local clients is small [e.g., 20 clients on CIFAR (Krizhevsky 2012)] and the client participation ratio is high (e.g., 40% on vision tasks and 100% on NLP tasks). With large amounts of local clients and stochastic client selection, FedDF faces several problems. FedED (Sui et al. 2020) extends FedDF for medical relation extraction. More practical scenes of utilizing FedDF are also studied, e.g., the resource-aware scenes (Yu et al. 2022).

2.4 Other related works

Learning from multiple source domains (Yao and Doretto 2010) is also related to TFL. The former does not consider the privacy protection policies and could send out source data or source models to facilitate the learning process of the target domain. The fundamental problem in these works is how to measure the transferability (Tong et al. 2021; Li et al. 2022) between source domains and the target domain. TFL considers data privacy protection, making the learning process more challenging. In TFL, we aim to simultaneously tackle the data heterogeneity problem and make better predictions for the to-be-inferred data.

3 Preliminaries

In this section, we first detail the setting and goal of TFL. Then, we introduce FedAvg (McMahan et al. 2017)/FedDF (Lin et al. 2020) and their drawbacks in TFL.

3.1 Transductive federated learning (TFL)

TFL also follows the parameter server (Li et al. 2013) architecture, and assumes training data are decentralized on local clients while the server could previously access the to-be-inferred data. Mathematically, we have K clients and each client owns a unique data distribution $\mathcal{D}_k = p_k(\mathbf{x}, y) = p_k(\mathbf{x})p_k(y|\mathbf{x}), k \in [K]$. We denote the observed samples as $\{(\mathbf{x}_{k,i}, y_{k,i})\}_{i=1}^{n_k}$, where n_k is the number of training samples on kth client. The total number of training samples from all clients is $N = \sum_{k=1}^{K} n_k$. In TFL, we assume the server owns an unlabeled set $\{\mathbf{x}_j\}_{j=1}^{M} \sim p_g(\mathbf{x})$ with M samples to be predicted. The goal of TFL is to make good predictions on the test set via collaborating with these K clients without transmitting clients' data. *Generally, we consider the data distribution of the test data (i.e., $p_g(\cdot)$) does not diverge a lot from the data distribution if all clients' data are centralized (i.e., $\frac{1}{K} \sum_{k=1}^{K} p_k(\cdot)$). We also consider the opposite case in Sect. 5.4 (i.e., cross-domain TFL).*

3.2 Federated averaging (FedAvg)

FedAvg (McMahan et al. 2017) takes T communication rounds of local and global procedures to collaborate with local clients. During local procedures, a small fraction (i.e. $R \in [0, 1]$) of clients S_t download the global model from server and update it on their own local data for E epochs. We denote the global model parameters in tth round as θ_t, and the updated model on kth client is $\theta_{t,k}$. During the global procedure, the server collects the updated models and takes a simple parameter averaging process as $\theta_{t+1} \leftarrow \frac{1}{|S_t|} \sum_{k \in S_t} \theta_{t,k}$. Faced with heterogeneous data, the local model update incurs large gradient variance and weight divergence (Zhao et al. 2018). In the following, we denote $f_k(\mathbf{x}; \theta_k)$ or f_k as the prediction function of the kth local model that outputs the "logits" for C classes, while $f_g(\mathbf{x}; \theta)$ or f_g as the prediction function of the aggregated model. We sometimes omit the communication round index t for

simplification. We use $q_k(y|\mathbf{x};\theta_k) = \sigma(f_k(\mathbf{x};\theta_k))$ to denote the predicted class probability distribution based on the kth local model, where $\sigma(\cdot)$ is the softmax operator. Similarly, $q_g(y|\mathbf{x};\theta)$ denotes the predicted probability of the aggregated model. Notably, $q(\cdot;\theta)$ denotes predicted probabilities while $p(\cdot)$ denotes the oracle ones.

3.3 Non-I.I.D. data

Because users' data is generated from different contexts, the data across federated clients is usually Non-I.I.D.. For experimental studies, previous works distribute a definite public data set (e.g., MNIST (Lecun et al. 1998), CIFAR (Krizhevsky 2012)) onto K clients according to various split strategies. In classification tasks with C classes, two commonly utilized ways are "*split by label*" and "*split by dirichlet*". The former assumes each client could only observe $\overline{C}$ classes while other $C - \overline{C}$ classes are not accessible (McMahan et al. 2017; Zhao et al. 2018; Li and Zhan 2021). Although some classes are missing, the observed classes are almost balanced. A smaller $\overline{C}$ corresponds to more serious Non-I.I.D. data. The latter samples a class distribution from the Dirichlet distribution $p_k(y) \sim \text{Dir}(\alpha)$ for each client (Hsieh et al. 2020; Lin et al. 2020), where α controls the Non-I.I.D. level, and a smaller α corresponds to a more Non-I.I.D. scene. After determining local clients' class distributions, training data are accordingly allocated to these clients for distributed training. We study both cases in this paper and show the split distributions with $K = 5$ clients in Fig. 2. *These two cases cover both challenges caused by data heterogeneity, data imbalance, amount skew, and missing classes, which are sufficient to verify the effectiveness of proposed FL methods.* Aside from these constructed Non-I.I.D. scenes split by classes, we also consider benchmarks split by users in experimental studies.

We implement FedAvg on several Non-I.I.D. cases via a one-shot FL similar to Guha et al. (2019) with only one communication round. Specifically, we first pretrain a global model on the centralized training set for r_0 steps and denote the
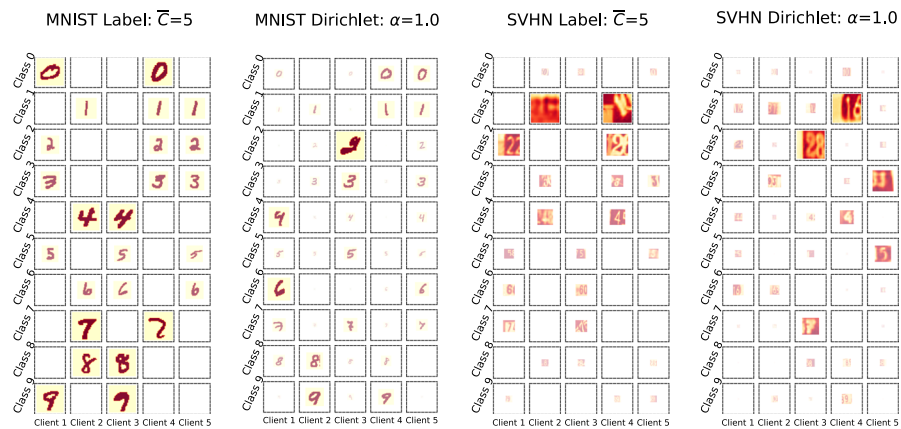

Fig. 2 Split data distributions with $K = 5$ clients. We split MNIST and SVHN via both "split by label" ($\overline{C} = 5$) and "split by dirichlet" ($\alpha = 1.0$). Darker colors and larger sizes mean more samples

obtained parameters as θ_0. We use θ_0 as initializations for both centralized training and decentralized training. For the former, we continually update θ_0 on the centralized training set for 50 SGD steps, and denote the obtained centralized model as θ_{Cen}. Then, we use θ_0 as global parameters and distribute it onto $K = 10$ clients constructed via the aforementioned split ways. We update θ_0 separately on these 10 clients for 50 steps, and denote the updated models as $\{\theta_{\text{Dec},k}\}_{k=1}^{K}$. In FedAvg, these updated models will be averaged on the server as the aggregated model, i.e., θ_{Agg}. We plot the extracted features of θ_0, θ_{Cen}, and θ_{Agg} under various Non-I.I.D. levels. For MNIST, we set the dimension of the final classification layer as 2 and plot the feature scatters. For SVHN, we extract hidden features and then utilize T-SNE (van der 2013) to obtain the 2-dimensional scatters. The figures are plotted in Fig. 3 where clusters with different colors represent different classes. The decentralized I.I.D. scenes (i.e., $\overline{C} = 10$, $\alpha = 10.0$) tend to perform better than centralized training because the former uses 10× training samples (10 clients). However, Non-I.I.D. data (i.e., $\overline{C} = 3$, $\alpha = 0.1$) experiences performance degradation and the features are less discriminative. In many FL theoretical analyses, local gradient variance among clients is always assumed to be bounded (Shamir et al. 2014; Li et al. 2020; Karimireddy et al. 2020), i.e., $E_k\left[\|\nabla_{\theta_k} f_k(\mathbf{x};\theta_k)\|^2\right] \leq \delta, \forall \mathbf{x}$. Intuitively, smaller gradient dissimilarity corresponds to better performances and faster convergence. We calculate the gradient variance as $\frac{1}{K}\sum_{k=1}^{K}\|\theta_{\text{Dec},k} - \theta_{\text{Agg}}\|^2$. Furthermore, the weight divergence proposed in Zhao et al. (2018) could also reflect the impact of Non-I.I.D. data, i.e., $\frac{\|\theta_{\text{Agg}} - \theta_{\text{Cen}}\|^2}{\|\theta_{\text{Cen}}\|^2}$. We calculate these two statistical measures under three Non-I.I.D. levels and plot the bars in Fig. 3, where Non-I.I.D. scenes really lead to larger local gradient variance and weight divergence. These findings conform to previous studies (Zhao et al. 2018; Li et al. 2020; Li and Zhan 2021). Additionally and originally, we also investigate the performance gap between Non-I.I.D. and I.I.D. training along with the quality of model initialization, i.e., varying pre-training steps to obtain different θ_0. We plot the performances of θ_0, θ_{Cen}, and θ_{Agg} under various Non-I.I.D. levels in rightmost of Fig. 3. The gap is significantly large when the initialization
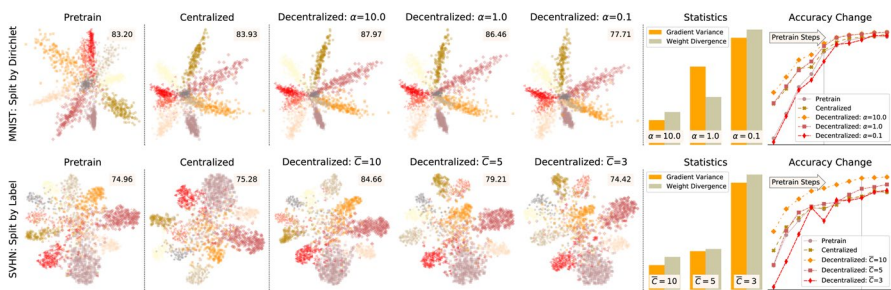

Fig. 3 Performance degradation of FedAvg under Non-I.I.D. scenes. The two rows show "split by dirichlet" on MNIST and "split by label" on SVHN, respectively. In each row, the left five figures show the extracted features and test accuracies (top-right numbers) of the pre-trained model, centralized model, and decentralized model under three levels of Non-I.I.D. data. The bars show two measures to evaluate the divergence of distributed training and centralized training. The rightmost shows the accuracy change of these five models with respect to pre-training steps

model is worse, while it reduces a lot with θ_0 becoming better. *This observation inspires us that solving the Non-I.I.D. problem in the beginning communication rounds of FL could be more valuable to accelerate training.* To be brief, Non-I.I.D. data lead to performance degradation of FedAvg.

3.4 Federated ensemble distillation (FedDF)

FedAvg takes an inductive manner and does not make utilization of the available test data in TFL. FedDF (Lin et al. 2020) could use ensemble distillation (Hinton et al. 2015) to fine-tune the aggregated model on the unlabeled test data in TFL. Mathematically, instead of simply averaging parameters as done in FedAvg, FedDF takes additional distillation steps to update the global aggregated model as follows:

$$\mathcal{L}_{\mathrm{KL},j} = \mathrm{KL}\left(\underbrace{\sigma\left(\frac{1}{|S_t|}\sum_{k\in S_t}f_k(\mathbf{x};\theta_k)\right)}_{\text{Distillation Targets}}, \sigma(f(\mathbf{x};\theta_{j-1}))\right), \tag{1}$$

$$\theta_j \leftarrow \theta_{j-1} - \eta\nabla_{\theta_{j-1}}E_{\mathbf{x}\sim p_g(\mathbf{x})}\left[\mathcal{L}_{\mathrm{KL},j}\right], \tag{2}$$

where θ_j is the aggregated model after the jth distillation step. KL denotes KL-divergence usually used in knowledge distillation (Hinton et al. 2015). The used $\mathbf{x}$ is originally obtained from a relevant public data set in FedDF, while we could directly sample $\mathbf{x}$ from the pre-available test data, i.e. $\sim p_g(\mathbf{x})$, in TFL.

FedDF significantly depends on the ensemble quality of local models, which is named as distillation targets in this paper (Eq. 1). The verified FL scenes in FedDF are cross-silo ones (Kairouz et al. 2019), where the number of clients is small (e.g., 20 clients) and clients' participation is stable (e.g., 40% or 100% client participation ratio). Furthermore, as declared in FedDF, local clients should undertake more local training steps (e.g., 40 or up to 160 epochs) to obtain ensemble models with enough diversity. These conditions may be too rigorous for some edge devices with unstable communication or limited computation. We consider a larger number of local clients (e.g., 100, 1000) and a smaller client participation ratio (e.g., 10%, 1%) in this paper. We run FedDF on decentralized SVHN as an example and plot the results in Fig. 4. We record test accuracies of the aggregated model, the ensemble of local updated models via "AvgLogi", and the distilled model obtained via Eqs. 1 and 2. Clearly, the "AvgLogi" distillation does not improve the aggregated model, and leads to training instability. Therefore, directly applying FedDF to TFL scenes seems to encounter some issues. We attribute the ineffectiveness of FedDF under these scenes to two reasons: *varying magnitudes* and *improper distillation*. We will detail these in the next section. *The essence of FedDF inspires us to propose more effective techniques to refine the inaccurate aggregated model.*
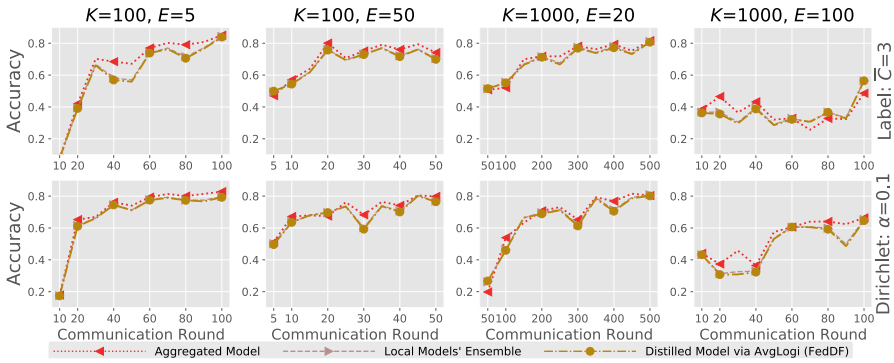
Fig. 4 Performance degradation of FedDF with larger number of clients (e.g., $K = 100, 1000$) and lower stochastic participation ratios (e.g., 10%, 1%). Rows show cases split by "label" and "dirichlet". E denotes the number of local training epochs

4 Proposed methods

In this section, we introduce our proposed methods. We follow FedDF (Lin et al. 2020) and polish it to be broadly applicable to TFL under more settings. Specifically, we propose Model refinery for Transductive Federated learning (MrTF) containing three modules: (1) stabilized teachers; (2) rectified distillation; (3) clustered label refinery.

4.1 Stabilized teachers

FedDF (Lin et al. 2020) takes "AvgLogi" to generate the distillation targets, i.e.,

$$\overline{q}_{\mathrm{AL}}(y|\mathbf{x}) = \sigma\left(\sum_k w_k f_k(\mathbf{x};\theta_k)\right), \tag{3}$$

while we consider another one via "AvgProb" as follows:

$$\overline{q}_{\mathrm{AP}}(y|\mathbf{x}) = \sum_k w_k \sigma\left(f_k(\mathbf{x};\theta_k)\right), \tag{4}$$

where we add weights for each client $w_k \geq 0$ satisfying $\sum_k w_k = 1$, and temporarily omit the client selection process for simplification (i.e., the $|S_t|$ in Eq. 1). We calculate the sensitivity of the targets $\overline{q}_{\star,c}$, $c \in [C]$, $\star \in \{\mathrm{AL}, \mathrm{AP}\}$ with respect to the local model parameters θ_k via calculating the gradients:

$$\frac{\partial \overline{q}_{\star,c}}{\partial \theta_k} = w_k J_{\star,c}(y|\mathbf{x})\left(\frac{\partial f_{k,c}}{\partial \theta_k} - \sum_j J_{\star,j}(y|\mathbf{x})\frac{\partial f_{k,j}}{\partial \theta_k}\right), \tag{5}$$

where $J_{\mathrm{AL}}(y|\mathbf{x}) = \overline{q}_{\mathrm{AL}}(y|\mathbf{x})$, $J_{\mathrm{AP}}(y|\mathbf{x}) = q_k(y|\mathbf{x};\theta_k)$. Obviously, the sensitivity is partially determined by the absolute value of the predicted probabilities $J_{\star}(y|\mathbf{x})$. *This implies that large probabilities make the distillation process sensitive to local*

models, while moderate prediction results are more stable. This also consists with some previous distillation research that find tolerant teachers will educate better students (Yang et al. 2019). Actually, we find that local updated models could generate "logits" with varying magnitudes on the same class, making "AvgLogi" suffer from large variance, and the predicted probabilities vary significantly across classes. We will show observations in experiments (Sect. 5.1).

To further reduce the "logits" variance and the sensitivity, we also normalize the "logits" before calculating probabilities in Eq. 4 as follows:

$$q_k(y|\mathbf{x};\theta_k) = \sigma\left(\tau * \frac{f_k(\mathbf{x};\theta_k)}{\text{std}\left(\{f_{k,c}(\mathbf{x};\theta_k)\}_{\mathbf{x}\sim p_g(\mathbf{x}),c\in[C]}\right)}\right), \quad (6)$$

$$\overline{q}_{AP}(y|\mathbf{x}) = \sum_k w_k q_k(y|\mathbf{x};\theta_k), \quad (7)$$

where $\text{std}(\{\cdot\})$ calculates the standard deviation of a set of values, i.e., all "logit" values of all classes on all test samples. τ is the temperature that controls the entropy and we use $\tau = 4.0$. *This normalization process could generate magnitude-invariant distillation targets among local models, which are more robust to averaging.*

4.2 Rectified distillation

From another aspect, the distillation in FedDF aims to optimize:

$$\min_\theta E_{\mathbf{x}\sim p_g(\mathbf{x})}\left[-\sum_{c=1}^C \overline{q}(y=c|\mathbf{x})\log q_g(y=c|\mathbf{x};\theta)\right], \quad (8)$$

where we use $\overline{q}(y|\mathbf{x}) = \overline{q}_{AP}(y|\mathbf{x})$ in Eq. 7 without any more consideration of "AvgLogi". This is just the KL-divergence in Eq. 1. Then, we rewrite the distillation process as:

$$\min_\theta E_{\mathbf{x}}\left[-\sum_{c=1}^C\left[\sum_{k\in S}\frac{w_k}{\sum_{j\in S}w_j}q_k(y=c|\mathbf{x};\theta_k)\log q_g(y=c|\mathbf{x};\theta)\right]\right], \quad (9)$$

where we consider stochastic client participation (i.e., only $|S|$ clients) resulted from limited or unstable communication. The ideal optimization of θ should be minimizing $\text{KL}(p_g(y|\mathbf{x}), q_g(y|\mathbf{x};\theta))$, $\forall \mathbf{x} \sim p_g(x)$. If we could guarantee $\sum_{k\in S}\frac{w_k}{\sum_{j\in S}w_j}q_k(y|\mathbf{x};\theta_k)$ approximates $p_g(y|\mathbf{x})$, the distillation process is unbiased and beneficial. This condition could be basically met in TFL if at least one of the following satisfies: (1) the clients' data distributions are the same with the global one, i.e., the I.I.D. case; (2) full or higher client participation in Non-I.I.D. case. The latter one explains why FedDF (Lin et al. 2020) is useful in cross-silo FL scenes. However, with a smaller set of participating clients, and supposing only the kth client is selected as an

extreme case, we actually minimize $\text{KL}(q_k(y|\mathbf{x};\theta_k), q_g(y|\mathbf{x};\theta))$. Because $q_k(y|\mathbf{x};\theta_k)$ is fitted to $p_k(y|\mathbf{x})$ and $p_k(y|\mathbf{x}) \propto p_k(y)p_k(\mathbf{x}|y)$, the distillation implicitly biases the global model θ to the kth client's prior distribution $p_k(y)$. Similarly, with a set of clients S, the aggregated model will be updated towards $\sum_{k \in S} \frac{w_k}{\sum_{j \in S} w_j} p_k(y)$. *In Non-I.I.D. cases, $\sum_{k \in S} \frac{w_k}{\sum_{j \in S} w_j} p_k(y)$ is not guaranteed to cover proper probabilities for all classes and experiences high variance with smaller set of S.* For example, we have $C = 4$ classes and select $S = \{1, 2\}$ with $p_1(y) = [0.5, 0.5, 0.0, 0.0]$ and $p_2(y) = [0.5, 0.0, 0.5, 0.0]$. We use uniform weights. Then the distillation process will be biased towards the distribution $[0.5, 0.25, 0.25, 0.0]$, which brings a negative transfer to the fourth class. We will verify this more in Sect. 5.1.

We propose two techniques to rectify the distillation targets. The first one is *enlarging the ensemble*. The initial global model and the aggregated model in the tth round are θ_t and θ_{t+1}, respectively, while the collected local models are $\{\theta_{t,k}\}_{k \in S_t}$. We use all of these models to generate distillation targets. Considering the aggregated models may perform worse in the beginning, we set a lower weight for them at previous communication rounds and gradually increase the weight. The second technique considers *a certain local model could only perform well on a portion of test data*. For example, if a local model only or majorly observes dogs and cats during local training, it could not teach or negatively teach the aggregated model to identify cars. We propose using the predicted entropy to measure how confident is the local model on the predicted sample, i.e., $e_{t,k}(\mathbf{x}) = -\sum_{c=1}^{C} q_k(y = c|\mathbf{x};\theta_{t,k}) \log q_k(y = c|\mathbf{x};\theta_{t,k})$. A smaller entropy corresponds to more confidence, and we let this model contribute more to the distillation process on this sample. Mathematically, the proposed rectified distillation targets are formulated as:

$$
\begin{aligned}
\overline{q}_{\text{RAP},t}(y|\mathbf{x}) = u_t * &\left(\sum_{k \in S_t} \frac{w_{t,k}(\mathbf{x})}{\sum_{j \in S_t} w_{t,j}(\mathbf{x})} q_k(y|\mathbf{x};\theta_{t,k}) \right) \\
&+ \frac{1 - u_t}{2} * \underbrace{\left(q_g(y|\mathbf{x};\theta_t) + q_g(y|\mathbf{x};\theta_{t+1}) \right)}_{\text{Self Teaching}},
\end{aligned}
\tag{10}
$$

where utilizing $\{w_{t,k}(\mathbf{x})\}_{k \sim S_t} = \sigma\left(-1.0 * \{e_{t,k}(\mathbf{x})\}_{k \in S_t}\right)$ can choose appropriate local models for each test sample to generate distillation targets. $q_k(\cdot)$ and $q_g(\cdot)$ are calculated as in Eq. 6. u_t balances the influence of local and global models. We adjust u_t via $u_t = 0.25 + 0.75 * \left(\frac{1}{|S_t|} \sum_{k \in S_t} \mathcal{L}_{t,k}\right) / \log C$. $\mathcal{L}_{t,k}$ denotes the local cross-entropy loss. With the loss becoming smaller, the aggregated models usually perform better and we enhance their influences. *Notably, fusing the initial model θ_t (i.e., the aggregated model in previous round) and the aggregated model θ_{t+1} could work as temporal ensembling or self-teaching such as in* Laine and Aila (2017), Tarvainen and Valpola (2017), Li et al. (2021).

4.3 Clustered label refinery

The aforementioned two modules separately provide solutions for the problem of varying magnitudes and improper distillation in FedDF (Lin et al. 2020). Only with these two modules, we could already yield better performances compared with FedDF. However, we additionally introduce other techniques to further enhance the stability and quality of the distillation targets. We take advantage of deep clustering (Caron et al. 2018) to *consider feature structural information*. This technique has been verified beneficial in domain adaptation (Liang et al. 2020) and transductive few-shot learning (Liu et al. 2019). Formally, we denote the obtained distillation targets as $\overline{q}(y|\mathbf{x}) = \overline{q}_{\mathrm{RAP},t}$ (Eq. 10). We extract hidden feature representations via the aggregated global model θ_{t+1} and denote the features as $\{h(\mathbf{x})\}_{\mathbf{x} \sim p_{\mathrm{g}}(\mathbf{x})}$. Then we further improve the distillation targets:

$$
\mathbf{v}_c = \frac{E_{\mathbf{x} \sim p_{\mathrm{g}}(\mathbf{x})} \big[\overline{q}_c(y|\mathbf{x}) h(\mathbf{x})\big]}{E_{\mathbf{x} \sim p_{\mathrm{g}}(\mathbf{x})} \big[\overline{q}_c(y|\mathbf{x})\big]},
\tag{11}
$$

$$
\overline{q}(y|\mathbf{x}) = \sigma\big(\{-1.0 * \tau * D_f(h(\mathbf{x}), \mathbf{v}_c)\}_{c=1}^{C}\big),
\tag{12}
$$

where $D_f(\cdot, \cdot)$ is a distance metric and we use $D_f(\mathbf{x}_1, \mathbf{x}_2) = 1.0 - \frac{\mathbf{x}_1^T \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|}$. τ is the temperature which is also set as 4.0. The two steps in Eqs. 11 and 12 could be iterated for several steps as done in unsupervised clustering (Caron et al. 2018), while we only take one step and it is enough to generate better distillation targets. *Notably, the aggregated model could not extract discriminative features in the beginning, thus we omit this process in the first several rounds (e.g., 5).*

4.4 MrTF

With the three modules, we propose MrTF as follows. During the tth communication round, the local procedure is the same as FedAvg (McMahan et al. 2017), while the global procedure takes several steps: (1) collect θ_t, the updated models $\{\theta_{t,k}\}_{k \in S_t}$, and the aggregated model θ_{t+1}; (2) make predictions for the global test set using these models in Eq. 6; (3) rectify these predicted probabilities in Eq. 10; (4) generate distillation targets via considering feature clusters in Eqs. 11 and 12; (5) refine the aggregated model θ_{t+1} on the global test set in Eqs. 1 and 2 with the replaced distillation targets. The refined global model is then distributed onto another set of clients for the next round of learning. The procedure of MrTF is illustrated in Fig. 5. The upload and download process is the same as FedAvg. The proposed stabilized teachers, rectified distillation, and clustered label refinery are aimed at generating better distillation targets. The refined model could simultaneously tackle the data heterogeneity challenge across clients and fuse the structural information of the to-be-inferred data.
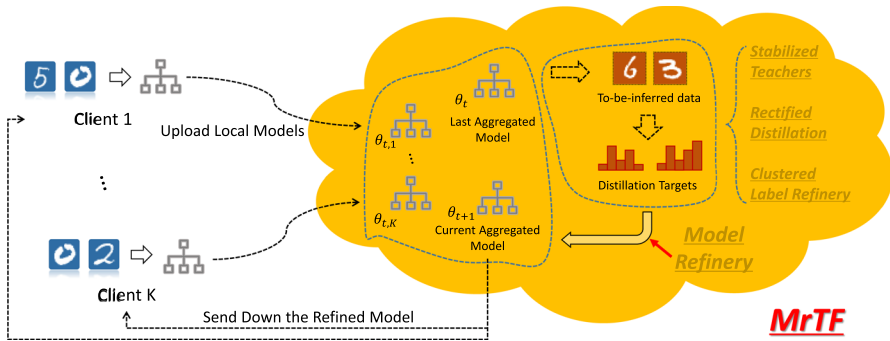
Fig. 5 The training procedure of the proposed MrTF. The proposed three techniques (i.e., stabilized teachers, rectified distillation, and clustered label refinery) could generate better distillation targets and facilitate the model refinery process

4.5 More discussion

We present more analysis of MrTF from other relevant aspects.

4.5.1 Individual distillation

"AvgProb" in Eq. 4 could bring another advantage that the distillation could be clearly decomposed into each client, which is more intuitive to analyze. Specifically, the loss in Eq. 9 could be viewed as $\sum_{k \in S} \frac{w_k}{\sum_{j \in S} w_j} \mathrm{KL}(q_k(y|\mathbf{x};\theta_k), q_g(y|\mathbf{x};\theta))$, where each client's model individually serves as a teacher. *Hence, we expect different teachers transfer different knowledge, i.e., their confident samples, implying the applied weights in Sect. 4.2 are more rational.*

4.5.2 Sensitivity to weights

We apply weights $w_k(\mathbf{x})$ in Eq. 10, and we could also add a uniform weight $w_k = 1/K$. If we do not use "AvgProb", directly utilizing "AvgLogi" has been verified sensitive to different weighting, shown in Fig. 6. Theoretically, the sensitivity of $\overline{q}_{\mathrm{AL}}$ with respect to w_k is relevant to the absolute value of $f_k(\mathbf{x};\theta_k)$ (Eq. 3), while in $\overline{q}_{\mathrm{AP}}$, it is relevant to $\sigma(f_k(x;\theta_k)) \in [0, 1]$. Obviously, the latter is more robust to the applied weights. This paves the foundation for adding two-level weights in rectified distillation (Sect. 4.2).

4.5.3 Self teaching

In the module of rectified distillation (Sect. 4.2), we add global aggregated models into the ensemble. We could decompose Eq. 10 into three parts: (1) the first is distilling local models' ability to the aggregated model; (2) the second is like $\mathrm{KD}(\theta_t, \theta_{t+1})$,

which utilizes historical prediction to supervise the current learning; (3) the third part is KD($\theta_{t+1}, \theta_{t+1}$), which is similar to self-teaching. KD($\cdot, \cdot$) denotes the knowledge distillation process.

5 Experiments

We use datasets from: (a) digits recognition: MNIST (Lecun et al. 1998), MNISTm (Ganin and Lempitsky 2015), SVHN (Netzer et al. 2011; b) image classification: CIFAR10/100 (Krizhevsky 2012), recommended by FedML (He et al. 2020; c) FeMnist and Shakespeare, recommended by LEAF (Caldas et al. 2018). Datasets in (a) and (b) are commonly utilized as benchmarks in centralized training. In our work, we split the corresponding training set onto K clients according to "split by label" with different $\overline{C}$ or "split by dirichlet" with different α. Smaller $\overline{C}$ and α lead to more Non-I.I.D. scenes, i.e., clients' data distributions differ a lot. Benchmarks in (c) provide a user list, and we construct Non-I.I.D. FL scenes via taking each user as an individual client. Specifically, Shakespeare is a dataset built from the Complete Works of William Shakespeare, which is originally used in FedAvg (McMahan et al. 2017). It is constructed by viewing each speaking role in each play as a different device, and the target is to predict the next character based on the previous characters. FeMnist is a task to classify the mixture of digits and characters, where data from each writer is considered as a client. These two benchmarks contain amounts of training samples and we only select 10% data for training. We list statistics of these benchmarks in Table 1 including: (1) the total amount of training samples of all clients (N); (2) the total number of test samples on the server (M); (3) the number of classes (C); (4) the number of clients (K); (5) the number of training samples of each client on average ($\overline{N_k}$); (6) the number of observed classes (i.e., at least 5 training samples) of each client on average when split by label ($\overline{C} = 3$) or dirichlet ($\alpha = 0.1$), denoted as $\overline{C_k}$ separated by "|".

For different datasets, we use corresponding deep neural networks, including: (1) MLPNet for MNIST with three layers, the hidden size of each hidden layer is 1024,

Table 1 Statistics of utilized datasets. We take $K = 100$ for an example

	N	M	C	K	$\overline{N_k}$	$\overline{C_k}$
MNIST (Lecun et al. 1998)	55k	10k	10	100	550	3.0 | 3.6
MNISTm (Ganin and Lempitsky 2015)	55k	10k	10	100	550	3.0 | 3.6
SVHN (Netzer et al. 2011)	73k	26k	10	100	730	3.0 | 3.9
CIFAR10 (Krizhevsky 2012)	50k	10k	10	100	500	3.0 | 3.6
CIFAR100 (Krizhevsky 2012)	50k	10k	100	100	500	30 | 17.5
FeMnist (Caldas et al. 2018)	85k	16k	62	359	236	15.3 | 14.4
Shakespeare (Caldas et al. 2018)	437k	84k	80	112	3.9k	28.7 | 29.2

and the last layer's size is 2 for visualization in Fig. 3 and 128 for performance comparisons; (2) LeNet (Lecun et al. 1998) for MNISTm; (3) ConvNet for SVHN as used in FedAvg (McMahan et al. 2017), we use T-SNE (van der 2013) for visualization in Fig. 3; (4) VGG8 (Simonyan and Zisserman 2015) for CIFAR10/100 with 5 convolution layers and 3 fully-connected layers; (5) ResNet8/20 (He et al. 2016) for CIFAR100; (6) FeCNN for FeMnist as used in LEAF (Caldas et al. 2018; 7) CharLSTM for Shakespeare as used in FedAvg (McMahan et al. 2017). For our proposed MrTF, we extract features for further label refinery as introduced in Sect. 4.3. For MLPNet and CharLSTM, we utilize the last hidden layer's output as features; for convolution networks, we use the flattened convolution features.

In TFL, the number of clients K, the client participation ratio R, the split parameters $\overline{C}$ and α determine a FL scene. Usually, K is large in FL, and R could be small due to limited or unstable communication. $\overline{C}$ and α are introduced to split the centralized training data for simulating a decentralized setting. We investigate $K = 100, 1000$, $R = 10\%, 1\%$ in our experiments. We also investigate several data split ways, e.g., $\overline{C} = 5, 3$ for $C = 10$, $\alpha = 1.0, 0.1$. Smaller $\overline{C}$ or α corresponds to more Non-I.I.D. scenes. Other important hyper-parameters include the number of global communication rounds T and the local training epochs E. We also study our method on various settings of T and E. We use SGD with a momentum of 0.9 as the local optimizer. For digits recognition scenes, we vary learning rate in $\{0.1, 0.05, 0.01\}$ and report the best one for comparison; for CIFAR scenes, we vary learning rate in $\{0.05, 0.03, 0.01\}$; for FeMnist, we use 0.004; for Shakespeare, we use 1.47. For digits and CIFAR scenes, we use a batch size of 64; for FeMnist and Shakespeare, we use 10. We use Adam with a learning rate 0.0003 as the global optimizer in FedDF and MrTF (Ours) and take 500 distillation steps.

5.1 Demo analysis

We first verify the success of the first two modules in MrTF, which are proposed to tackle the varying magnitudes and improper distillation drawbacks in FedDF (Lin et al. 2020). We experiment on MNIST with three clients and each client could only observe two classes. We init a global model and distribute it to the three clients. After local training, we use these three local models to predict on the global test set, recording the accuracy and each instance's "logits" and "probs". We average the class "logits" or "probs" across test samples for better presentation. Because the global test set is uniformly distributed across 10 classes, and *we expect the average results of both "logits" and "probs" are also uniform*. The results are shown in Fig. 6.

First, the top three figures show the results of each local model. The accuracies are low, i.e., 17.6%, 19.2%, 20.4%. The reasons are intuitive: they are trained with only 2 classes, while the global test set contains 10 classes. The "logits" across clients vary greatly, with the largest ranging from 9.0 to 20.0, while the corresponding "probs" are limited to [0, 0.5]. If we uniformly ($w_k = 1/K$) average "logits" and "probs" of three local models for each test instance as done in Eqs. 3 and 4, the results are shown at the bottom left of Fig. 6. Because the 9th class (showed in red)
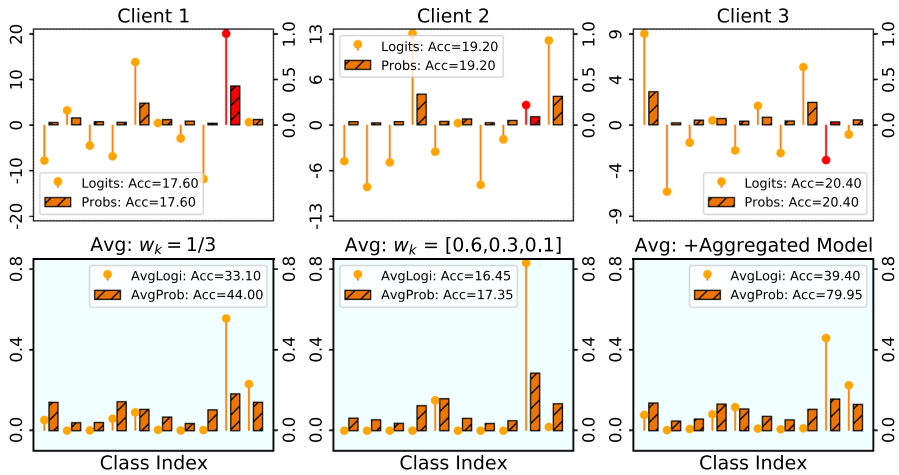
Fig. 6 Comparisons of "AvgLogi" and "AvgProb" across three clients on MNIST, each client only observes 2 classes. The top shows the instance-averaged "logits" and "probs" on global test set of each local model. The bottom shows the distillation targets generated via: uniform averaging, non-uniform averaging, averaging after adding aggregated models (Color figure online)

generally has large "logits" (i.e., around 20.0) predicted by the first local model, it dominates the $\sigma(\cdot)$ operation and makes "AvgLogi" output much higher probabilities on the 9th class. However, using "AvgProb" leads to smoother class probabilities and the test accuracy improves from 33.1 to 44.0%. If we apply $w_1 = 0.6, w_2 = 0.3$, $w_3 = 0.1$ for averaging, the results of "AvgLogi" are worse as shown in the bottom middle of Fig. 6 (the 9th stem is higher). However, "AvgProb" performs more stably and the class probabilities are more uniform. *These observations show that replacing "AvgLogi" with "AvgProb" could really mitigate the problem of the varying magnitude, leading to moderate teachers and better ensemble performances.*

From another aspect, because these three clients only observe at most 6 classes in total, some unseen classes' "logits" will be inaccurate. Illustrated in Fig. 6, some classes' probabilities become zero. That is, the stochastic client participation will lead to inaccurate distillation targets, and directly using "AvgLogi" or "AvgProb" for distillation is improper. Instead, we fuse the global aggregated models and rectify the probabilities as done in Sect. 4.2. Then the results shown at the bottom right of Fig. 6 are better. That is, the probabilities of "AvgLogi" and "AvgProb" become more smooth and the test accuracies are improved to 39.4% and 79.9%, respectively. All observations verify the rationality and effectiveness of our solutions in Sects. 4.1 and 4.2.

5.2 Performance comparisons

We compare MrTF with FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020), FedMMD (Yao et al. 2018), FedOpt (Reddi et al. 2021), Scaffold (Karimireddy et al. 2020), and FedDF (Lin et al. 2020). The first five algorithms do not access the global test set during training, while they utilize various techniques to solve Non-I.I.D.

problems. FedDF utilizes "AvgLogi" to refine the aggregated model, which is the most similar to ours. Details of these algorithms are presented as follows.

- *FedAvg* McMahan et al. (2017): the most standard FL algorithm that utilizes parameter averaging for model aggregation.
- *FedProx* Li et al. (2020): introduces a proximal term during local procedures to constrain the model parameters' update.
- *FedMMD* Yao et al. (2018): introduces the discrepancy minimizing optimization (i.e., MMD) in local procedures and regularizes the local model not diverge a lot from the global model too much.
- *FedOpt* Reddi et al. (2021): updates the global model via momentum or adaptive optimization techniques to stabilize the global model's update.
- *Scaffold* Karimireddy et al. (2020): points out the local update will diverge from the global direction and utilizes control variates to reduce local gradient variance.
- *FedDF* Lin et al. (2020): uses local models' ensemble, i.e., "AvgLogi", to finetune the global model on a relevant public data set.

5.2.1 Part I

We first study on MNIST, MNISTm, SVHN, and CIFAR10, which have 10 classes to identify. We construct four FL scenes for each dataset via "split by label" with $\overline{C} \in \{5, 3\}$ and "split by dirichlet" with $\alpha \in \{1.0, 0.1\}$. We take $K = 100$ clients and only select $R = 10\%$ clients in each communication round. We update $E = 3$ epochs for each client during local procedures and take $T = 200, 1500$ communication
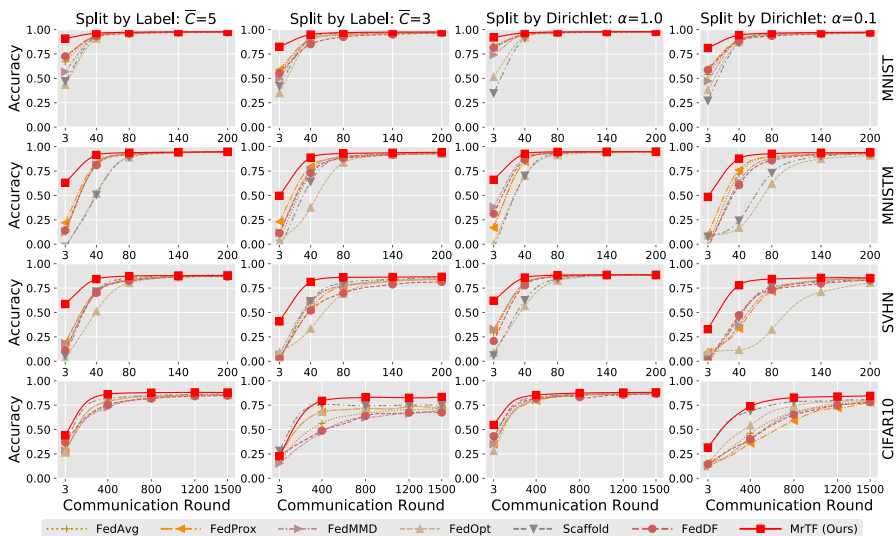

Fig. 7 Performance comparisons on several FL scenes. Row shows each dataset and column shows each data split way

rounds for digits and CIFAR scenes, respectively. The results are shown in Fig. 7, where MrTF converges faster and performs better on all scenes. First, MrTF could surpass other methods by a large margin especially in the beginning, verifying that learning from local models' ensemble significantly helps and lays the foundation for subsequent improvements. This conforms to the observation in Sect. 3.3. Some compared algorithms could only improve FL on certain scenes. For example, Scaffold performs better than others on CIFAR10, while worse on other datasets.

5.2.2 Part II

Then, we vary the utilized networks and compare the performances on CIFAR100 using VGG8 (Simonyan and Zisserman 2015) and ResNet8/20 (He et al. 2016). We take $K = 100$ clients and $R = 10\%$. The results are shown in Fig. 8. For each scene, we run 500 communication rounds and each client takes on $E = 20$ epochs. In some cases, MrTF performs worse than Scaffold, attributed to the control variates used in
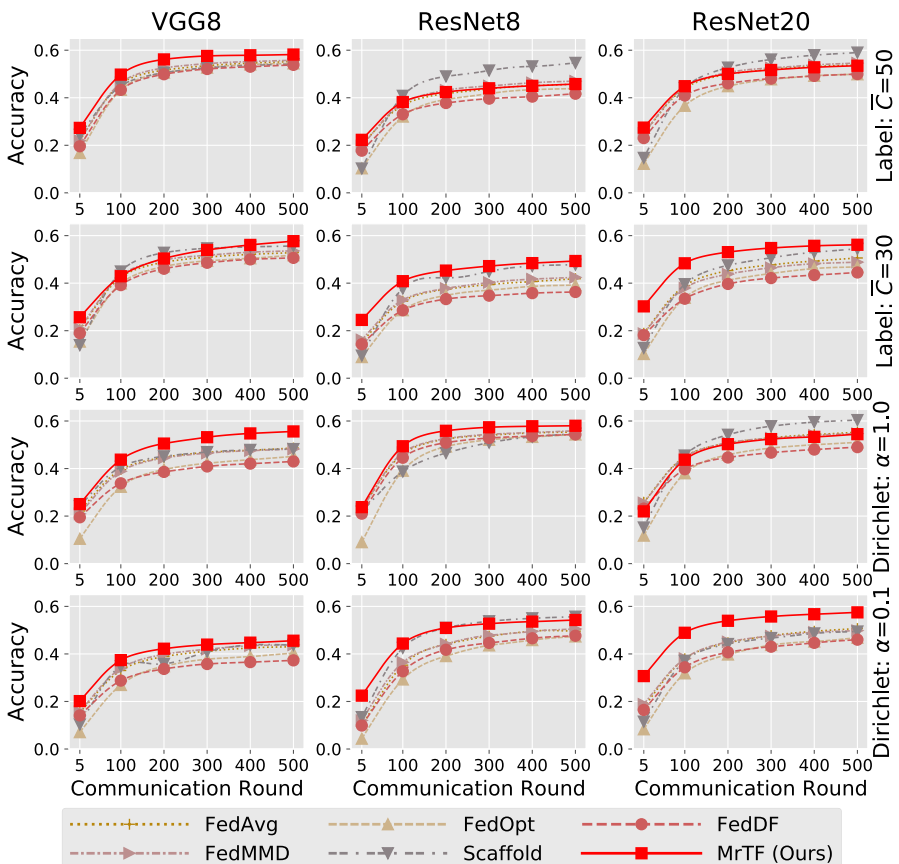
Fig. 8 Performance comparisons on CIFAR100 based on VGG8 and ResNet8/20. Row shows FL scene with different data split ways

Scaffold. However, MrTF could obtain better results on most of the cases, especially in more Non-I.I.D. scenes, i.e., $\overline{C} = 30$ and $\alpha = 0.1$ (the 2nd and 4th row in Fig. 8). Because we have 100 classes, the possibility that participating clients cannot cover all classes greatly increases, making FedDF ineffective.

5.2.3 Part III

We also investigate our method on LEAF (Caldas et al. 2018) benchmarks, i.e., FeMnist and Shakespeare. These two benchmarks are split by users, where the distribution skew dominates the Non-I.I.D. (Kairouz et al. 2019) problem. We show the results in Fig. 9. Our proposed MrTF could still show effectiveness towards other methods. Although Scaffold could achieve faster convergence in the beginning, the performance degrades a lot with a larger communication round. The training instability limits the application of Scaffold to TFL. Compared with this, our proposed MrTF could achieve the Scaffold's best performance and is more stable.

5.2.4 Part IV

We then majorly compare with FedAvg and FedDF on various scenes. Specifically, we vary $K \in \{100, 1000\}$ to investigate large amounts of clients. For $K = 100$, we take $R = 10\%$ to select only 10 clients in each round and each client updates $E \in \{2, 10, 50\}$ local epochs; for $K = 1000$, we use $R = 1\%$ and $E \in \{5, 50, 100\}$. We experiment on SVHN and CIFAR10 under two split strategies, i.e., "split by label" with $\overline{C} = 5$ and "split by dirichlet" with $\alpha = 1.0$. The results are plotted in Fig. 10. MrTF can basically surpass FedAvg and FedDF on all scenes. Additionally, MrTF behaves more stably even with larger number of clients (e.g., $K = 1000$), contributed to the stabilized teachers and rectified distillation.

5.3 Ablation studies

Our proposed MrTF contains three modules: (1) we use "AvgProb" in Eq. 7 instead of "AvgLogi" to obtain stabilized teachers; (2) we fuse aggregated models into local
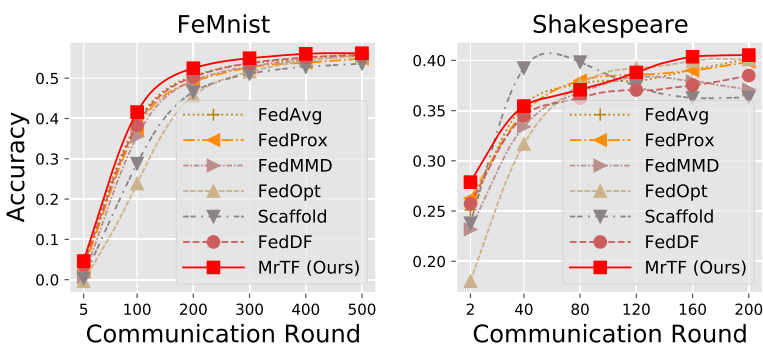

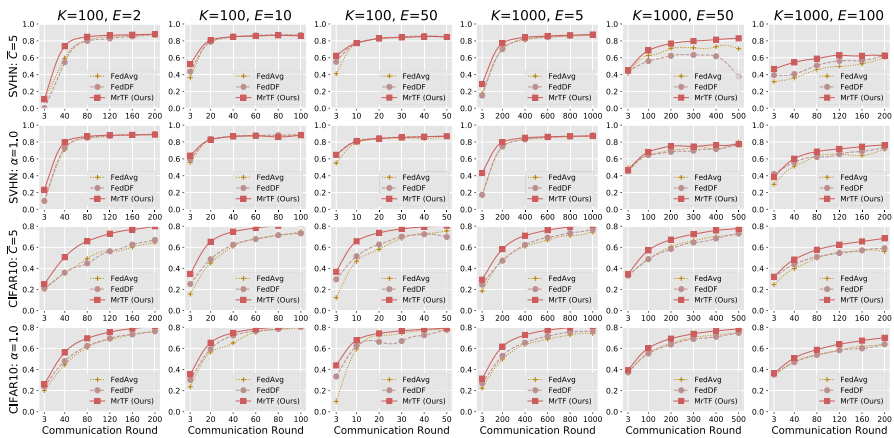**Fig. 9** Performance comparisons on LEAF benchmarks

Fig. 10 Comparison results with FedAvg and FedDF on more FL scenes. Row shows dataset and corresponding split strategy, and column shows the number of clients K and the number of local training epochs E

models and apply two-level weights for rectified distillation; (3) we additionally take clustering techniques to refine the distillation targets. We incrementally add these modules for ablation studies. We denote these three components as "ST", "RD", and "CLR". Correspondingly, we compare performances of: (1) simple parameter averaging without any refinery (i.e., FedAvg); (2) averaging logits (i.e., "AvgLogi" in FedDF); (3) ST; (4) ST + RD; (5) ST + RD + CLR (i.e., proposed MrTF). We compare them on SVHN and CIFAR10 under various FL scenes, and the results could be found in Fig. 11. For each scene, we run 50 communication rounds. "AvgLogi" is not stable and sometimes surpasses parameter averaging while sometimes does not. Only using the "AvgProb" in Eq. 7 (i.e., ST) could already yield notable performances, while fusing RD and CLR could lead to higher results.

5.4 More studies: cross-domain TFL

In some cases, although a party could collaborate with other parties to help infer the handy unlabeled data via the proposed TFL framework, the distribution of the
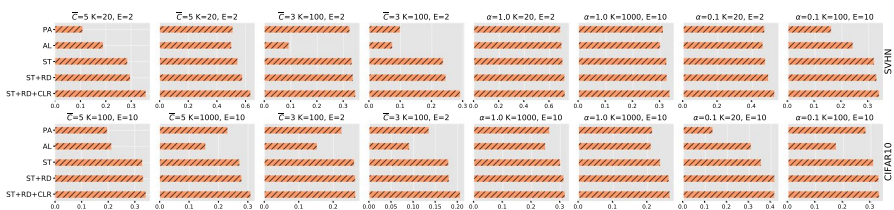

Fig. 11 Ablation studies of the modules in MrTF. Each row shows a data set. Each column shows the split strategy ($\overline{C} \in [5, 3]$, $\alpha \in [1.0, 0.1]$) and corresponding K, E. The five bars refer to test accuracy of: (1) PA (parameter averaging in FedAvg); (2) AL ("AvgLogi" in FedDF); (3) ST (Stabilized Teachers, Sect. 4.1); (4) ST + RD (Rectified Distillation, Sect. 4.2); (5) ST + RD + CLR (Clustered Label Refinery, Sect. 4.3) (MrTF)

Table 2 Performance comparisons in cross-domain TFL

	SVHN→ MNISTm		MNISTm→ SVHN	
	$\alpha=1.0$	$\alpha=0.1$	$\alpha=1.0$	$\alpha=0.1$
FedAvg (McMahan et al. 2017)	40.26	34.52	32.09	31.06
FedDF (Lin et al. 2020)	41.90	35.91	33.06	28.84
MrTF (Ours)	44.78	41.70	35.32	34.55

Table 3 Performances when adding differential privacy

	$\sigma = 0.0$	$\sigma = 0.001$	$\sigma = 0.01$	$\sigma = 0.1$
FedAvg (McMahan et al. 2017)	79.46	76.98	69.08	38.27
MrTF (Ours)	82.31	81.76	76.54	47.11

unlabeled data may also be heterogeneous from others. We call this case cross-domain TFL, which is similar to the scene studied in Peng et al. (2020); Feng et al. (2021). These works only consider several heterogeneous domains (e.g., 5), which are more similar to domain adaptation under privacy protection (Long et al. 2015; Liang et al. 2020). That is, they do not consider some other challenges in our work, i.e., stochastic client participation, low-shot training samples, class imbalance, etc. In cross-domain TFL, we have to simultaneously tackle these challenges aside from Non-I.I.D. data and cross-domain knowledge transfer. We preliminarily apply MrTF to this scene. Specifically, we split SVHN (MNISTm) data across $K = 100$ clients with $\alpha \in \{1.0, 0.1\}$. The server aims to make predictions for MNISTm (SVHN). In each round, we select 10 clients and each client takes on 5 epochs. We run 200 communication rounds and report the final accuracies averaged by 5 independent experiments. We compare with FedAvg and FedDF. Results are listed in Table 2. MrTF could still surpass FedAvg and FedDF by a significant margin even in cross-domain TFL. However, the overall cross-domain transfer performance is still lower compared with in-domain learning, which means that more advanced domain adaptaion (Long et al. 2015) techniques should be considered for cross-domain TFL.

5.5 More studies: privacy protection

FedAvg could only provide basic privacy protection for users, while some advanced attacks could still break privacy via inverting local gradients (Zhu et al. 2019; Geiping et al. 2020). Hence, techniques such as differential privacy (Abadi et al. 2016) should be considered for stricter privacy protections. To guarantee (ϵ, δ)-DP in FL, gradient clipping is applied to local model updates, and gaussian noises $\mathcal{N}(0, \sigma^2)$ are added before being sent to the server. With added noise, the aggregated model will be more inaccurate. However, we expect our model refinery process could mitigate the performance degradation. We experiment on CIFAR10 with $\alpha = 1.0, K = 100, E = 5, T = 200$. We use VGG8 and add noise

$\sigma \in \{0.0, 0.001, 0.01, 0.1\}$. We report the results of FedAvg and MrTF in Table 3. With higher noise, FedAvg's performance degrades seriously while MrTF could maintain a better prediction.

5.6 Limitations and future work

Our proposed MrTF is a novel and practical solution to the introduced real-world scenario in that a newly-established pilot project needs to build a machine-learning model with the help of other isolated parties. However, MrTF does not consider the existing models of these parties and trains local models from scratch, making the convergence slower. Utilizing the available pre-trained models and accelerating the training process may be interesting for future work.

6 Conclusion

We consider transductive federated learning (TFL), where the server owns to-be-referred data while the training data are distributed across other parties. We in-depth analyze some existing FL works and point out their drawbacks. As an alternative, we propose MrTF with three modules, i.e., stabilized teachers, rectified distillation, and clustered label refinery, to refine the global aggregated model and make predictions in a transductive manner. Our proposed method shows superiorities towards compared methods on various investigated scenes.

References

Abadi M, Chu A, Goodfellow IJ, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the ACM SIGSAC conference on computer and communications security, pp 308–318

Afonin A, Karimireddy SP (2022) Towards model agnostic federated learning using knowledge distillation. In: The Tenth international conference on learning representations

Caldas S, Wu P, Li T, Konecný J, McMahan HB, Smith V, Talwalkar A (2018) LEAF: a benchmark for federated settings. arxiv:abs/1812.01097

Caron M, Bojanowski P, Joulin A, Douze M (2018) Deep clustering for unsupervised learning of visual features. In: Computer Vision: ECCV 2018, pp 139–156

Chen Y, Wang G, Dong S (2002) Learning with progressive transductive support vector machine. In: Proceedings of the IEEE international conference on data mining, pp 67–74

Feng H, You Z, Chen M, Zhang T, Zhu M, Wu F, Wu C, Chen W (2021) KD3A: unsupervised multi-source decentralized domain adaptation via knowledge distillation. In: Proceedings of the 38th international conference on machine learning, pp 3274–3283

Ganin Y, Lempitsky VS (2015) Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd international conference on machine learning, pp 1180–1189

Geiping J, Bauermeister H, Dröge H, Moeller M (2020) Inverting gradients: how easy is it to break privacy in federated learning? In: Advances in neural information processing systems. 33

Guha N, Talwalkar A, Smith V (2019) One-shot federated learning. arxiv:abs/1902.11175

He C, Li S, So J, Zhang M, Wang H, Wang X, Vepakomma P, Singh A, Qiu H, Shen L, Zhao P, Kang Y, Liu Y, Raskar R, Yang Q, Annavaram M, Avestimehr S (2020) Fedml: A research library and benchmark for federated machine learning. arxiv:abs/2007.13518

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on computer vision and pattern recognition, pp. 770–778

Hinton GE, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arxiv:abs/1503.02531

Hsieh K, Phanishayee A, Mutlu O, Gibbons PB (2020) The non-iid data quagmire of decentralized machine learning. In: Proceedings of the 37th international conference on machine learning, pp. 4387–4398

Jeong E, Oh S, Kim H, Park J, Bennis M, Kim S (2018) Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arxiv:abs/1811.11479

Jeong W, Yoon J, Yang E, Hwang SJ (2021) Federated semi-supervised learning with inter-client consistency & disjoint learning. In: 9th International conference on learning representations

Kairouz P, McMahan HB, et al (2019) Advances and open problems in federated learning. arxiv:abs/1912.04977

Karimireddy SP, Kale S, Mohri M, Reddi SJ, Stich SU, Suresh AT (2020)SCAFFOLD: stochastic controlled averaging for federated learning. In: Proceedings of the 37th international conference on machine learning, pp. 5132–5143

Krizhevsky A (2012) Learning multiple layers of features from tiny images

Laine S, Aila T (2017) Temporal ensembling for semi-supervised learning. In: 5th International conference on learning representations

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Liang J, Hu D, Feng J (2020) Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: Proceedings of the 37th international conference on machine learning, pp. 6028–6039

Li X, Huang K, Yang W, Wang S, Zhang Z (2020) On the convergence of fedavg on non-iid data. In: 8th International conference on learning representations

Lin T, Kong L, Stich SU, Jaggi M (2020) Ensemble distillation for robust model fusion in federated learning. In: Advances in neural information processing systems. 33

Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V (2020)Federated optimization in heterogeneous networks. In: Proceedings of machine learning and systems

Li X, Tang J, Song S, Li B, Li Y, Shao Y, Gan L, Zhan D (2022) Avoid overfitting user specific information in federated keyword spotting. In: Proceedings of the 23rd annual conference of the international speech communication association, pp. 3869–3873

Liu Y, Lee J, Park M, Kim S, Yang E, Hwang SJ, Yang Y (2019) Learning to propagate labels: transductive propagation network for few-shot learning. In: 7th International conference on learning representations

Li D, Wang J (2019) Fedmd: Heterogenous federated learning via model distillation. arxiv:abs/1910.03581

Li X, Wang Y, Gan L, Zhan D (2022) Exploring transferability measures and domain selection in cross-domain slot filling. In: IEEE International conference on acoustics, speech and signal processing, pp. 3758–3762

Li X, Xu Y, Song S, Li B, Li Y, Shao Y, Zhan D (2022) Federated learning with position-aware neurons. In: IEEE/CVF conference on computer vision and pattern recognition, pp. 10072–10081

Li X, Zhan D (2021) Fedrs: federated learning with restricted softmax for label distribution non-iid data. In: The 27th ACM SIGKDD conference on knowledge discovery and data mining, pp. 995–1005

Li X, Zhan D, Shao Y, Li B, Song S (2021) Fedphp: Federated personalization with inherited private models. In: Proceedings of the machine learning and knowledge discovery in databases: European conference, pp. 587–602

Li M, Zhou L, Yang Z, Li A, Xia F, Andersen DG, Smola A (2013) Parameter server for distributed machine learning. In: Big Learning NeurIPS workshop. 6:2

Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd international conference on machine learning, pp. 97–105

Long Z, Wang J, Wang Y, Xiao H, Ma F (2021) Fedcon: a contrastive framework for federated semi-supervised learning. arxiv:abs/2109.04533

McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence and statistics. pp. 1273–1282

Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng A (2011) Reading digits in natural images with unsupervised feature learning

Peng X, Huang Z, Zhu Y, Saenko K (2020) Federated adversarial domain adaptation. In: 8th International conference on learning representations

Reddi SJ, Charles Z, Zaheer M, Garrett Z, Rush K, Konečný J, Kumar S, McMahan HB (2021) Adaptive federated optimization. In: 9th International conference on learning representations

Rohrbach M, Ebert S, Schiele B (2013) Transfer learning in a transductive setting. Adv Neural Inf Process Syst 26:46–54

Shamir O, Srebro N, Zhang T (2014) Communication-efficient distributed optimization using an approximate newton-type method. In: Proceedings of the 31th International conference on machine learning. pp. 1000–1008

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International conference on learning representations

Sui D, Chen Y, Zhao J, Jia Y, Xie Y, Sun W (2020) Feded: Federated learning via ensemble distillation for medical relation extraction. In: Proceedings of the conference on empirical methods in natural language processing. pp. 2118–2128

Tarvainen A, Valpola H (2017) Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: 5th International conference on learning representations

Tong X, Xu X, Huang S, Zheng L (2021) A mathematical framework for quantifying transferability in multi-source transfer learning. Adv Neural Inf Process Syst 34:26103–26116

van der Maaten L (2013) Barnes-hut-sne. In: 1st International conference on learning representations

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008

Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 10(2):12–11219

Yang Y, Sun Z-Q, Zhu H, Fu Y, Zhou Y, Xiong H, Yang J (2021) Learning adaptive embedding considering incremental class. IEEE Trans Knowledge Data Eng

Yang C, Xie L, Qiao S, Yuille AL (2019) Training deep neural networks in generations: A more tolerant teacher educates better students. In: Proceedings of the 33rd AAAI conference on artificial intelligence, pp. 5628–5635

Yao Y, Doretto G (2010) Boosting for transfer learning with multiple sources. In: The Twenty-Third IEEE conference on computer vision and pattern recognition, pp. 1855–1862

Yao X, Huang C, Sun L (2018) Two-stream federated learning: Reduce the communication costs. In: IEEE visual communications and image processing, pp. 1–4

Yu S, Qian W, Jannesari A (2022) Resource-aware federated learning using knowledge extraction and multi-model fusion. arxiv:abs/2208.07978

Zhang L, Shen L, Ding L, Tao D, Duan L (2022) Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In: IEEE/CVF conference on computer vision and pattern recognition, pp. 10164–10173

Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V (2018) Federated learning with non-iid data. arxiv:abs/1806.00582

Zhu L, Liu Z, Han S (2019) Deep leakage from gradients. Adv Neural Inf Process Syst 32:14747–14756