International Journal of Scientific Engineering and Science
Volume 8, Issue 4, pp. 20-24, 2024. ISSN (Online): 2456-7361

Model Heterogeneous Federated Learning with Noisy Data

Peng Yu[1]
[1]College of Information Engineering, Nanjing University of Finance & Economics, Nanjing, Jiangsu, 210023
1565517911@qq.com

***Abstract*—*Each client designs its own model independently, making the task even more difficult. Existing algorithms are not efficient in solving the problem of varying noise in local clients, which is caused by the difficulty of data labeling and hitchhiking clients. In this paper, we address the challenging problem of federated learning with noisy and heterogeneous clients. We propose a new solution, Federated Classifier Jointing (FedClassJoint), which simultaneously handles label noise and performs federated learning in a single framework. The deep neural network used for the supervised learning task consists of a feature extractor layer and a classifier layer. Additionally, we apply local feature representation learning to stabilize the decision boundary and improve the local feature extraction capability of the client. FedClassJoint is characterized by three aspects: (1) efficient communication between heterogeneous models, achieved by requiring the client to communicate with only a few fully connected layers; (2) reduction of negative impact caused by internal label noise through the use of contrastive regularization loss function (CCTR). (3) To address noisy feedback from other participants, we have designed a new client confidence reweighting scheme. This scheme adaptively assigns appropriate weights to each client classifier during the collaborative learning phase. The classifier weights are then aggregated into a decision boundary protocol on the feature space, resulting in a powerful global classifier. Our approach has been extensively tested and has proven effective in minimizing the negative impact of various noise rates and types in both homogeneous and heterogeneous federated learning settings. It consistently outperforms existing methods.*

***Keywords*—*Heterogeneous federated learning, label nosie, classifier, model heterogeneity.*

I. INTRODUCTION

Local clients such as mobile devices or entire organizations typically have limited private data and limited generalizability. Therefore, centralized learning of public models using private data from all clients would greatly improve performance. However, the existence of data silos and data privacy prevents us from using traditional centralized learning in real-world applications [1]. To address these challenges, McMahan proposed federated learning (FL) [2]. Federated learning is a distributed machine learning framework that allows multiple clients and a global server to train by exchanging knowledge from local training and the data itself. Clients never share private data with the server, ensuring basic privacy. Starting with FedAvg [2], many studies have been proposed to improve the generalization performance of federated learning algorithms. However, because federated learning concentrates on improving global models, the performance of client models with local data distribution deteriorates. Therefore, the concept of personalized federated learning has been proposed. Its goal is to allow clients to collaboratively train personalized models while maintaining model performance on local data distributions. Therefore, many heterogeneous federated learning approaches have been proposed in order to perform federated learning with heterogeneous models [5-8]. FedDF [6] performs an integrated extraction using unlabeled data for each of the different model architectures. FedMD [5] is a framework based on knowledge distillation by means of class scores of the client models on a public dataset. These strategies mainly rely on a unified global consensus or shared model. However, one major limitation of learning global consensus is that clients cannot individually adjust their learning directions to accommodate differences between clients. In addition, building additional models will increase computational overhead, which affects efficiency and effectiveness. Therefore, it is a challenge to perform joint learning using heterogeneous clients without relying on global consensus or shared models.

The above methods mainly rely on the assumption that each client has a clean dataset, which cannot be satisfied in many practical applications. When the clients contain unavoidably noisy samples, existing joint learning methods are unable to eliminate the negative effects of labelling noise, and thus suffer significant performance degradation [8]. Since joint learning involves a large number of clients, the data in each client usually has different noise patterns. Typically, in practice, labelling noise is caused by two aspects: (1) The quality of the labelled data is affected by human subjective factors and due to the limited and scarce human expertise, which means that high-quality labelled data requires a high cost, which inevitably leads to some incorrect labelling. (2) In the federated learning framework, considering the issue of user fairness, there may be some free riders in the system who want to learn from the global model but do not want to provide useful information. So, some users are not willing to share their real information with other users and deliberately generate some false labels. To reduce the negative effects of labelling noise, existing methods [10-13] are usually developed for image classification tasks with a single model. These methods can be classified into four categories: label transfer matrix estimation [14-17], robust regularization [18], robust loss function design [20], and clean sample selection [22]. Under the joint learning framework, we expect each class of samples to be fully learned while avoiding overfitting noisy samples. Therefore, how to reduce the negative impact of internal labelling noise on the local model convergence during

All rights reserved

the local update phase is an important issue. In addition, how to reduce the negative and noisy effects of other clients in a collaborative learning framework is also an important issue. Existing approaches to address noise in machine learning can only eliminate the negative impact of internal model labeling noise, but not noise from other clients. Therefore, it is crucial to deal with noisy feedback from other noisy clients in a joint learning framework.

This paper presents solutions to FedClassJoint for federated learning with noisy and heterogeneous clients. Firstly, we propose locally noisy learning with contrast regularized loss functions to address the negative impact of internal model labeling noise. During the local learning phase, the contrast regularization function retains information related to the true labels and discards information related to the corrupted labels. (2) Client Confidence Re-weighting for External Noise. To reduce the impact of label noise resulting from feedback from other clients, we employ Client Confidence Re-weighting (CCR) in joint communication. No changes in content have been made. This technique reduces the contribution of noisy clients and unites the uniform client classifier weights to form a powerful global classifier via CCR. With a formal register and exact word choice, the language is objective, clear, and succinct. The content follows standard formatting guidelines and organization, with uniform citation and footnote styles. The grammar, spelling, and punctuation are correct.

The contributions of this paper and the proposed FedClassJoint are as follows: We further investigate the robust federated learning problem with noisy and heterogeneous clients. We adaptively adjust the weights of the classifiers for each client during the loss update by CCR to reduce the contribution of noisy clients and increase the contribution of clean clients, resulting in a robust global classifier. By combining classifier aggregation and local representation learning, we introduce FedClassJoint, a novel framework for personalized federation learning on heterogeneous models. We validate the proposed approach in various settings, including heterogeneous and homogeneous models with different noise types and noise rates. Experimental results show that FedClassJoint consistently works better than other methods.

II. RELATED WORK

2.1 Heterogeneous Federated Learning

Several studies have identified the possibility of joint learning approaches for heterogeneous models [24, 25]. Personalized joint learning approaches for heterogeneous models using knowledge transfer have also been proposed in the literature [26]. Knowledge refinement or transfer transfers learned information to models with different architectures, making them suitable for successful application in heterogeneous training. KT-pFL[27] is a knowledge transfer-based personalized federated learning algorithm for heterogeneous models that has achieved state-of-the-art performance. In this algorithm, a global server aggregates local soft predictions from clients on broadcast public data. The server then computes the knowledge transfer coefficient, which determines how much knowledge should be updated from one client to another. However, collecting more public data remains a major issue. It is important to ensure that public and private data have similar distributions. This can only be achieved if semantic information on private client data is available. Knowledge transfer-based algorithms require local models to be trained for multiple periods, which can be computationally expensive for clients that are often assumed to be edge devices. Furthermore, the computation of knowledge coefficients and their application to the local model necessitates additional computation on the client side. Previous studies have not addressed this issue, but FedZKT [28] resolves it by shifting computationally intensive tasks from the client to the global server through the aggregation of client model weights. This approach, however, sacrifices communication overhead. In summary, current knowledge transfer-based approaches either compromise client privacy, overburden edge computing resources, or require excessive communication. However, FedClassJoint exchanges only two fully connected layers without using auxiliary datasets. This promotes effective communication.

2.2 Label Noise Learning

In machine learning, several methods have been proposed to handle labeling noise. These methods can be classified into four main groups:

(1) Label transfer matrices. The primary concept is to estimate the probability of each labeling class flipping to another class. Sukhbaatar [14] introduced a noise layer to the network to align the network output with the noisy labeling distribution. Patrini [15] developed an end-to-end loss correction framework to enable the application of state-of-the-art noise estimation techniques to multi-class settings.

(2) Robust regularization. Robust regularization was employed to prevent models from overfitting noisy labels. Zhang [18] proposed Mixup, which trains convex combinations of sample pairs and their labels to regularize hybrid neural networks. Arpit[19] showed that regularization slows down the recall of noise while not affecting the learning of real data.

(3) Robust loss functions. Some methods achieve robust learning by using noise-tolerant loss functions. Rooyen [20] proposed a convex classification calibration loss which is robust to symmetric labeling noise. Ghosh[21] analyzed some widely used loss functions in deep learning and proved that MAE is robust to noise.

(4) Select potentially clean samples. These methods select clean samples from a noisy training dataset to use for learning, or reweight each sample. The core idea is to reduce the focus on noisy labeled samples in each iteration of training. In order to train two deep neural networks concurrently and choose data with possibly clean labels for cross-training, Han[22] developed Cooperative Teaching. Wei[23] proposed JoCoR, which uses joint regularization to compute the joint loss, and then selects small-loss samples to update the network parameters.

Previous methods to solve label noise are mainly in local learning. But in joint learning, the server does not have direct access to the client's private dataset. Different model architectures result in varying noise patterns and inconsistent

All rights reserved

Volume 8, Issue 4, pp. 20-24, 2024. ISSN (Online): 2456-7361

decision boundaries in model heterogeneity environments. How to effectively combine inconsistent decision boundaries is an important issue that will be explored in this paper.

III. METHODOLOGY

3.1 Definition of the problem

In joint learning, each K-client learns local model weights using local data containing noise to improve the performance of the global model. Thus, the goal of joint learning can be stated as follows:

$$\min_{w} \mathcal{L}(f(w:x),y) := \sum_{k=1}^{k} \frac{|D_k|}{|D|} \mathcal{L}(f(w:x),y) \quad (1)$$

Where $\mathcal{L}$ is the loss function, f is the model architecture, w is the global model weight parameter, w_k is the model weight parameter of client k, $D = (x, y)$ denotes the oracle dataset, i.e., when aggregating client data, where x is the input data and y is the corresponding label.

However, personalized federated learning for heterogeneous models aims to minimize the average client loss over the local data distribution. The goal of personalized federated learning for client loss $\mathcal{L}_k$, data (x_k, y_k) and model architecture f_k can be stated as follows:

$$\min_{w_1,\dots,w_K} \sum_{k=1}^{k} \frac{|D_k|}{|D|} \mathcal{L}_k(f_k(w_k; x_k), y_k). \quad (2)$$

3.2 Local noise Learning

In fact, since we only get a noisy dataset, we do not know if the labels are clean or not. Therefore, simply minimizing (1) leads to performance degradation. To see this, note that (1) activates only when $\{x_k, y_k\}$ = 1. Thus, two representations from different classes will be pulled together when noisy labels are present. Since the deep network first fits examples with clean labels and the probabilistic output of these examples is higher than that of examples with corrupted labels, it helps in representation learning only in the early stages. After that time, the examples with corrupted labels will dominate the learning process, as the magnitude of the gradient from the correct contrast pair overwhelms the magnitude of the gradient from the incorrect contrast pair. In particular, given two clean examples xi, xj, yi = yj and one mislabeled example xm, ym = yi = yj, after an early phase, the deep network begins to fit the mislabeled data. At this point, the incorrect comparison pairs (xi, xm) and (xj, xm) are incorrectly pulled together, and they impair representation learning over the correct pair (xi, xj).

To address this problem, we propose the following regularization function to avoid the negative impact of incorrect contrast pairs:

$$\tilde{\mathcal{L}}_{ctr}(x_i, x_j) = \left(log\left(1 - \langle \tilde{q}_i, \tilde{z}_j \rangle\right) + log\left(1 - \langle \tilde{q}_j, \tilde{z}_i \rangle\right)\right) \{p_i \top p_j \geq \tau\} (3)$$

The purpose of (3) remains to learn similar representations of data with the same true label. Furthermore, gradient analysis of (3) shows that the L2 paradigm gradient increases when $\tilde{q}_i$ and $\tilde{q}_j$ converge. Our proposed regularization function is not dominated by the gradient of the wrong pair, and the model does not overfit clean examples even if the gradient of (3) for the correct pair is larger than that of the wrong pair. The magnitude of its gradient can be viewed as the strength of pulling clean samples of the same class closer together, which is not directly related to overfitting clean samples. In addition, we use a separate identical linear layer at the top of each client representation as a classifier, so as long as the gradient of the classification loss relative to the parameters in the linear layer is not large on the clean examples, the model will not overfit them.

Finally, the overall objective function is given:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \tilde{\mathcal{L}}_{ctr} \quad (4)$$

where $\tilde{\mathcal{L}}_{ctr}$ serves as the contrast regularization (CTRR) of the representation and λ controls the strength of the regularization.

To balance local knowledge with knowledge learned from other clients, we set up a local learning phase. Clients will update the local model using their own private dataset to prevent local knowledge from being forgotten. During the training iterations, labeling noise can cause the model to update in the wrong direction and eventually fail to converge. To avoid this result, we use (4) to calculate the loss between the pseudo-label predicted by the model and the corresponding given label. Customers utilize (4) to update the model while enhancing local knowledge, avoiding overfitting of noisy labels and facilitating adequate learning.

3.3 Client Confidence Re-weighting

We propose the Client Confidence Re-weighting (CCR) method to reduce the adverse effects of label noise from other clients during the collaborative learning phase. CCR can personalize the weights of each client during the communication process, reduce the contribution of noisy clients, and pay more attention to the clients with clean datasets and efficient models. And the weights of each client after CCR reconstruction are utilized to weight its classifier, resulting in a powerful global classifier. To estimate the label quality, (4) is used to compute the loss between the predicted output of a local model for a private noisy dataset and a given label. To quantify the learning efficiency, we compute the (4) drop rate for each round of iterations. the Loss drop rate returned by the client in T rounds of iterations is denoted as ΔL. Specifically, the loss decrease rate can reflect the learning efficiency of the model to some extent. Then we simply use the Loss drop rate to quantify the learning efficiency of the client C_k. By quantifying the label quality of the private dataset and the learning efficiency of the local model, the confidence of each client is measured separately. In the collaborative learning phase, we reweight the confidence of each client so that the client can learn more from the quality clients and reduce the learning weight for the inferior clients, while assigning a larger weight to the classifiers of the quality clients and a smaller weight to the classifiers of the inferior clients. As the training iterates, each model is updated towards clean and efficient clients and a strong global classifier is formed.

http://ijses.com/
All rights reserved

International Journal of Scientific Engineering and Science
Volume 8, Issue 4, pp. 20-24, 2024. ISSN (Online): 2456-7361

3.4 Summary

The whole process is shown in Fig. 1. First, each client C_k updates the local model f_k and its classifiers with a private noise dataset to obtain a set of pre-trained models and their classifiers D_k. In collaborative learning, clients C_k utilize CCR to align the feedback distributions of other clients, and clients can then individually adjust their learning direction according to the differences in the models, instead of simply learning the global consensus. Therefore, to reduce the effect of local noise, we use CCTR in (4) to update the local model, and then calculate the client confidence based on label quality and model learning efficiency. When learning knowledge distributions from other clients, the client's confidence is reweighted to the participants and their classifiers. Through personalized weighting, the participation of noisy clients in the federated system is adjusted to form a powerful global classifier that avoids the effect of noise in the communication process.

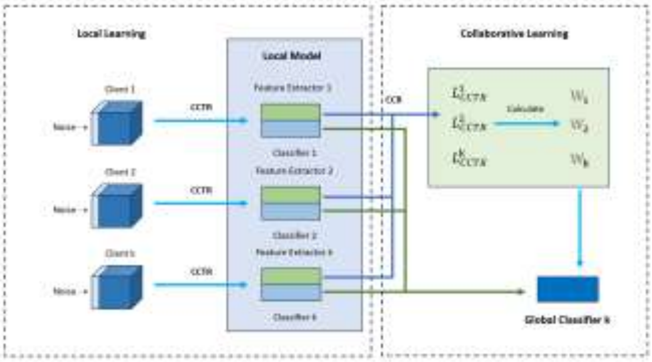

Fig. 1. Architecture diagram of FedClassJoint. Feature Extractor is the feature extractor and Classifier is the classifier. FedClassJoint aggregates client classifiers and builds a global classifier.

IV. EXPERIMENTS

A. Experimental settings

Datasets and models. Our experiments are conducted on two datasets, Cifar10 and Cifar100, which are widely used in the study of labeling noise. Here we set the public dataset on the server as a subset of Cifar100 and randomly partition Cifar10 into different clients as the private dataset.

In the heterogeneous model scenario, we assign four different networks ResNet10, ResNet12, ShuffleNet and Mobilenetv2 to each of the four clients. While in the homogeneous model scenario, the network of all four clients is set to ResNet12.

Noise type. We use the label transfer matrix M to add label noise to the dataset. Matrix M has two widely used structures:symmetric flip and pair flip. Symmetric flip means that the original class label will be flipped to any wrong class label with equal probability. For pair flipping, it means that the original class labels will only be flipped to very similar wrong classes.

Implementation details. The sizes of the private and public datasets are specified as Nk = 10,000 and N0 = 5,000, respectively. we perform Tc = 40 co-learning epochs for different models. We use the Adam optimizer with an initial learning rate of α = 0.001 and a batch size of 128. λ is set to 0.1 and η is set to 0.5. We select the noise rate μ = 0.1 and 0.2 because the main focus of this research is on the robustness of federated learning under noisy supervision. We next go over the results under both pair-flipping and symmetric flipping noise types. To generate noisy datasets, we invert 20% of the labels in the training dataset of Cifar10 to the wrong labels and keep the test dataset of Cifar10 unchanged to observe the robustness of the model. Results of the experiment

B. Results of the experiment

In the same setting, we compare with state-of-the-art heterogeneous federated learning methods. The baseline is the method where the client trains a local model on a private dataset without federated learning. Therefore, the comparison of the two noise rates is shown in Tables 1 and 2. The experiments show that our proposed method outperforms the existing strategies under different noise conditions. In Pairflip noise, our proposed method FedClassJoint outperforms the current SOTA method RHFL by 0.49%, FedDF by 1.70%, and FedMD by 1.86%. In Symflip FedClassJoint is 0.36% higher than RHFL, 3.84% higher than FedDF, and 3.58% higher than FedMD. FedClassJoint has good results in either noise scenario.

TABLE I. Experimental results under Pairflip noise.

Method	**Pairflip**				
	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	77.98	76.75	66.89	74.33	73.99
FedMD	74.98	76.89	67.10	76.64	73.90
FedDF	76.26	75.51	**68.41**	76.04	74.06
RHFL	78.76	**79.72**	67.99	74.20	75.17
FedClassJoint	**79.90**	77.68	68.11	**77.43**	**75.76**

TABLE II. Experimental results under Symflip noise.

Method	**Symflip**				
	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	76.20	76.05	64.96	74.31	72.88
FedMD	73.23	73.66	67.72	75.54	72.54
FedDF	72.07	75.18	67.38	74.47	72.28
RHFL	77.71	**79.04**	70.54	76.20	75.76
FedClassJoint	**78.84**	76.98	**71.11**	**77.53**	**76.12**

V. CONCLUSION

In this paper, we study the problem of how to perform robust federated learning under noisy heterogeneous clients. To solve this problem, a new FedClassJoint solution is proposed. In order to avoid overfitting of each model to the noise during the local learning process, the CCTR loss is used to update the local models. For the noise feedback from other participants, we use a flexible reweighting method, CCR, which effectively avoids overlearning from noisy clients, and at the same time unites the classifiers of all participating clients to form a powerful global classifier, realizing noise-resistant federated learning. A large number of experiments demonstrate the effectiveness in our approach.

REFERENCES

[1] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cum-mings, Advances and open problems in federated learn-ing. arXiv preprint arXiv:1912.04977, 2019.

All rights reserved

[2] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication- efficient learning of deep networks from decentralized data.In AISTATS, pages 1273–1282, 2017.

[3] Shaoxiong Ji, Teemu Saravirta, Shirui Pan, Guodong Long, and Anwar Walid.2021. Emerging Trends in Federated Learning: From Model Fusion to FederatedX Learning. arXiv preprint arXiv:2102.12920 (2021).

[4] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, MehdiBennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cor-mode, Rachel Cummings, 2021. Advances and Open Problems in FederatedLearning. Foundations and Trends® in Machine Learning 14, 1–2 (2021), 1–210.

[5] Daliang Li and Junpu Wang. Fedmd: Heterogenous feder-ated learning via model distillation. In NeurIPS Workshop,2019.

[6] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In NeurIPS, page 2351–2363, 2020.

[7] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. In NeurIPS Workshop, 2020.

[8] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-freeknowledge distillation for heterogeneous federated learning.In ICML, 2021.

[9] Li Li, Huazhu Fu, Bo Han, Cheng-Zhong Xu, and LingShao. Federated noisy client learning. arXiv preprint arXiv:2106.13239, 2021.

[10] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In NeurIPS, pages 8792–8802, 2018.

[11] Seunghan Yang, Hyoungseob Park, Junyoung Byun, and Changick Kim. Robust federated learning with noisy labels. arXiv preprint arXiv:2012.01700, 2020.

[12] Mang Ye and Pong C Y uen. Purifynet: A robust person re-identification model with noisy labels. IEEE TIFS, 2020.

[13] Mang Ye, He Li, Bo Du, Jianbing Shen, Ling Shao, andSteven CH Hoi. Collaborative refining for person re-identification with label noise. IEEE TIP, 2021.

[14] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In ICLR, 2015.

[15] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In CVPR, pages 1944–1952, 2017.

[16] Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W Tsang, and JunSun. Safeguarded dynamic label regression for noisy super-vision. In AAAI, pages 9103–9110, 2019.

[17] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In NeurIPS, pages 5841–5851, 2018.

[18] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, andDavid Lopez-Paz. mixup: Beyond empirical risk minimization. In ICLR, 2018.

[19] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Y oshua Bengio, A closer look at memorization in deep networks. In ICML, pages 233–242, 2017.

[20] Brendan V an Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: The im-portance of being unhinged. In NeurIPS, pages 10–18, 2015.

[21] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In AAAI, pages 1919–1925, 2017.

[22] Bo Han, Quanming Yao, Xingrui Y u, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with ex-tremely noisy labels. In NeurIPS, 2018.

[23] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In CVPR, pages 13726–13735, 2020.

[24] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In International Conference on Learning Representations (ICLR).

[25] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. Advances in Neural Information Processing Systems 33 (2020), 14068–14080.

[26] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous Federated Learning via Model Distillation. arXiv preprint arXiv:1910.03581 (2019).

[27] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. 2021. Parameterized Knowledge Transfer for Personalized Federated Learning. Advances in Neural Information Processing Systems 34 (2021).

[28] Lan Zhang and Xiaoyong Yuan. 2021. FedZKT: Zero-Shot Knowledge Transfer towards Heterogeneous On-Device Models in Federated Learning. arXiv preprint arXiv:2109.03775 (2021).

[29] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2021. FedProto: Federated prototype learning over heterogeneous devices. The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI) (2021).

All rights reserved