MGFL: Multi-granularity Federated Learning in Edge Computing Systems

Shangxuan Cai, Yunfeng Zhao, Zhicheng Liu, Chao Qiu, Xiaofei Wang(✉), and Qinghua Hu

College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
{shangxuancai,yfzhao97,liuzhicheng,chao.qiu,xiaofeiwang,huqinghua}@tju.edu.cn

Abstract. As a promising machine learning framework in the big data era, federated learning (FL) allows multiple mobile devices to collaboratively train a model without transmitting raw data, thus has attracted widespread attention in both academia and industry. Considering that heterogeneous mobile devices with limited resources and data diversity are bound to impact the actual performance of some training nodes. However, conventional FL could not support collaborative training with multi-granularity neural networks. To this end, we propose multi-granularity federated learning (MGFL) that contains two mechanisms serving for same-granularity FL and cross-granularity FL. MGFL customizes a personalized model for each device by designing a divergence-based similarity measurement method in same-granularity FL. Further, it adjusts the empirical risk loss function to break the restriction of cross-granularity FL. Experimental evaluations demonstrate the positive guidance of the fine-granularity model to the coarse-granularity model, which significantly improves the performance of the coarse-granularity model. Besides, our method shows superiority on both independently identically distribution (IID) and non-IID data.

Keywords: Federated learning · Edge computing · Multi-granularity model · Personalized model

1 Introduction

The Internet of Things gets rapidly development since the increased demand for convenience in life. Its network applications, such as mobile heterogeneous electric vehicle networks in smart city [1] and wireless sensor and actuator networks [2], catalyzes enormous amounts of data from edge devices every second. Since these data contain rich and valuable information, they have attracted wide attention from researchers in recent years. Considering that these messages often need more complex calculations to decode the valuable parts of them, some practical technologies are applied to edge computing [3–5]. As an indispensable universal service for big data processing [6], deep learning (DL) promotes the intelligence

© Springer Nature Switzerland AG 2022
Y. Lai et al. (Eds.): ICA3PP 2021, LNCS 13155, pp. 549–563, 2022.
https://doi.org/10.1007/978-3-030-95384-3_34

of mobile devices under the influence of the popularity of deep neural network (DNN) models, which have become an essential part of edge computing systems.

However, there is generally insufficient data on one mobile device to sustain training a high-quality model. Data are often owned by different entities, which means that individual's data privacy is compromised if centralized improperly [7,8]. Thus, privacy and data integrity become essential points for application or data owners [9]. With this in mind, individuals are reluctant to interact and share data to train the model. Federated learning (FL) is proposed to alleviate privacy anxiety and achieve cooperative training among numerous edge devices (a.k.a. clients) [10]. In recent years, FL has drowned extensive discussions for its superior ability on data privacy protection. The general FL system permits all clients to keep their data locally and only upload parameters or gradients to collaboratively train models under the coordination of a central server.

Kairouz *et al.* [11] discussed the existing problems and future development of FL, in addition to which there are still many bottlenecks limiting the applications of FL in edge computing. Conventional FL only allows all clients to share one global model, which is not suitable for heterogeneous clients. One factor of heterogeneity is the diversity of data distribution that makes the aggregated model fail to apply to each participant. Another factor is the constrained resources, including storage, computing power, energy, and other such things. Those heterogeneous clients with constrained resources fail to train DL models ideally, resulting in each client being forced to get a model with low performance.

As coarse-granularity models typically require fewer training resources than fine-granularity models, a sensible approach is to run DL models with coarse granularity if the client does not impose a requirement for granularity. E.g., a fine-granularity model generally runs on resource-rich clients to identify cars, boats, men, women, cats, dogs, etc. In contrast, a coarse-granularity model runs on resource-constrained clients to identify coarse-granularity individuals, such as vehicles, people, animals, etc. The heterogeneous clients in edge computing systems have constrained resources and diverse data, leading to multi-granularity models.

As client diversity is not conducive to the regular cooperation among multiple clients, many researchers have attempted to solve this dilemma in FL. Some studies try to design a personalized model for each client [12] and address the challenge of different data distributions among clients. The works in [13,14] select relevant and irrelevant clients according to the data distributions, and works in [15,16] aggregate similar client models through the idea of clustering. Another work in [17] considers the condition of the heterogeneous resource and non-IID data in clients, aiming to solve the selection problem of clients. However, these works only focus on the differences in data samples and lack consideration of similar samples with different labels. The studies in [18,19] analyze the collaboration among multiple tasks, and the work in [20] further explores the non-IID data effect based on multi-task. However, they all ignore the situation of multi-granularity tasks. Although data labels are different among multi-granularity models, which is contrary to the principle of FL, the similarity among samples is the same as the original intention of FL that aims to make up for the lack of samples among clients.

To address the challenges above, we propose a novel FL method, named multi-granularity federated learning (MGFL), aiming to promote the cooperative training among models with different granularity. MGFL adopts the same-granularity FL to provide a personalized model for each client and adopts cross-granularity FL with adjusting the empirical risk loss function, which further improves the models' performance of coarse-granularity clients.

Our method focuses on the scenario of multi-granularity clients while performing significantly on non-IID data. The main contributions of this paper are summarized as follows:

- We propose a novel FL method named MGFL, which contains two mechanisms, i.e., same-granularity FL and cross-granularity FL, promoting FL among models of different granularity clients in edge computing systems.
- We design a divergence-based similarity measurement method to provide a personalized model for each client in same-granularity FL. Under the guidance of fine-granularity models, we improve models' accuracy of coarse-granularity clients by adjusting the empirical risk loss function in cross-granularity FL.
- We conduct extensive experiments demonstrating that MGFL is more effective for adaptive FL with multi-granularity clients. It shows better performance on both IID and non-IID data than other baselines and achieves great performance by accurately capturing relationships among clients.

The remainder of this paper is organized as follows. Section 2 shows the motivation of this work. Section 3 presents the system model and the proposed MGFL mechanism. Extensive performance evaluations are presented in Sect. 4. Section 5 concludes this paper.

2 Motivation

We compare the model differences between coarse-granularity and fine-granularity clients when performing image classification tasks. Since parameters are basic representations of the models, we Visualize the differences in models' image feature extraction capabilities with different granularity by comparing parameters. We choose two models with different granularity based on pre-trained Wide-Resnet [21] as observation objects and compare all layers except the final dense layer.

Let models A and B be trained under the CIFAR-100 dataset [22] with the same samples and different granularity labels, i.e., coarse-granularity and fine-granularity labels, respectively. We measure the distance between the output vectors of the two models at each layer using the Euclidean distance, which indicates the difference in the parameters of each layer. As shown in Fig. 1(a), there are significant differences in the convolutional layer parameters between the coarse-granularity and the fine-granularity model. In particular, as the network layers deepen, the difference in parameters between layers increases. It illustrates that there are significant differences between models with different granularity even trained with the same samples.

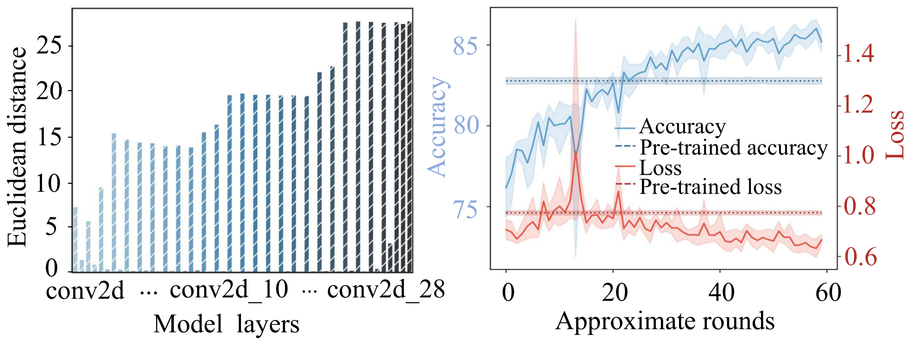

Fig. 1. Two key observations for motivating cross-granularity FL: (a) Euclidean distance of all layers between coarse-granularity model A and fine-granularity model B; (b) Model A's performance during the process of approximating to B's parameters.

Next, we attempt to make the coarse-granularity model A's parameters approximate to those of the fine-granularity model B and test the models' accuracy and loss after several rounds of modifications to the parameters. The results are shown in Fig. 1(b), where the dashed and solid lines represent the performance of the initial pre-trained and the approximating models, respectively. The performance of the coarse-granularity model gradually improves as the parameters continue to approach those of the fine-granularity model and eventually surpass the performance of the pre-trained model. The result show the positive guiding effect of fine-granularity models to coarse-granularity models in feature-extracting of convolution layers and lay the foundation of cross-granularity FL.

Based on the above two observations, we expect to overcome the limitation of conventional FL that only trains the same models and collaborates to train coarse-granularity and fine-granularity models with widely varying model parameters. On the other hand, we aim to give full play to the guidance of the fine-granularity model to the coarse-granularity model during FL to further improve the coarse-granularity model's performance. Therefore, we propose the MGFL method to satisfy the collaborative training among multi-granularity clients in edge computing systems.

3 System Model

In this section, we design MGFL to realize FL among multi-granularity clients containing similar model structures and samples.

3.1 The Multi-granularity Federated Learning Framework

As shown in Fig. 2, extensive clients at the edge are managed by server and divided into coarse-granularity clients $\mathcal{N} = \{1, 2, ..., N\}$ and fine-granularity clients $\mathcal{M} = \{1, 2, ..., M\}$. At the beginning of MGFL, each client sends its model weight as well as a part of its shared data and accuracy of recent rounds to the server. The aggregation phase is divided into two stages dominated by same-granularity FL and cross-granularity FL, respectively. In the early stage, the same granularity models are measured by a similarity method based on the Jensen-Shannon divergence (JS divergence) that urges same-granularity models to aggregate the new personalized models. In the later stage, the fine-granularity clients perform same-granularity FL while coarse-granularity clients perform cross-granularity FL with the guidance of fine-granularity clients. In cross-granularity FL, each coarse-granularity model selects the most relevant fine-granularity model as the target to guide it and then modify parameters with the help of fine-granularity one. The server sends personalized models back to each client, and clients will start the next round of local training. The entire training is iterated until each client receives the highest-performing model. The corresponding algorithm is shown in Algorithm 1. Assume that the system is in a quasi-static state during a round, which means no clients join or leave during a round.

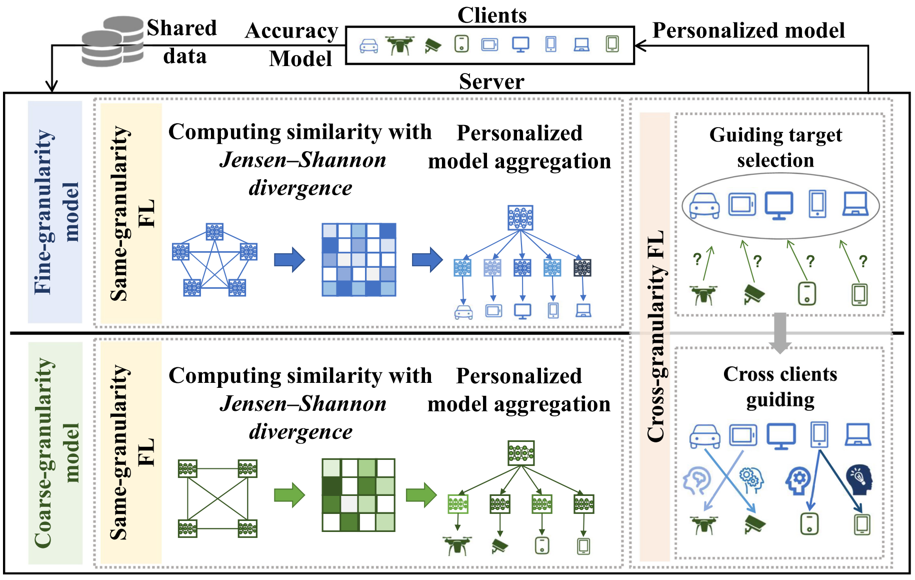

Fig. 2. The framework of MGFL.

3.2 Same-Granularity Federated Learning

One of the challenges of designing MGFL is the non-IID data kept locally by the client, which severely affects the quality of aggregation due to differences in sample classification. To decrease the impact caused by it and design personalized models on non-IID data [23], the early stage of MGFL captures the relationship among clients using a similarity-based method. By taking the coarse-granularity clients as an example, the problem at the early stage is formulated as follows:

$$\min_{W} \sum_{i=1}^{N} \frac{1}{N_i} \sum_{j=1}^{N_i} F_i(w_i, x_{i,j}, y_{i,j}), \tag{1}$$

where N is the number of total involved clients. N_i is the number of training data samples in client i. $(x_{i,j}, y_{i,j}) \in \mathbb{R}^E \times \mathbb{R}$ is the j-th training data pair of client i where E is the number of data features; $W = [w_1, ..., w_N]^{\mathrm{T}}$ is the weight matrix to be estimated consisting of all the local models' weight. F_i is the loss function of the i-th client model.

Local Model Computation: Each client i will parallelly optimize (e.g., using stochastic gradient descent methods) its model weight w_i based on its local data (x_i, y_i). In the t-th local iteration, each client computes its local update, i.e.,

$$w_i^t = w_i^t - \eta_l \nabla F_i(w_i^{t-1}), \tag{2}$$

where η_l is the predefined learning rate during local training.

Server Model Aggregation: After finishing local iterations, the client i will transmit its local model parameters w_i to the parameter server. There is a challenge when learning the aggregation relationship among clients using their model parameters at the server: how to quantify the similarity among models. Considering that the number of the last dense layer's parameters of the model in different granularity is generally different, the parameter server spontaneously distinguishes these models into two parts according to it. Same-granularity FL means the FL only processes clients in the same granularity. When the server receives models from clients, we endurance the distance of models using each client's shared dataset.

There are some classical methods used in computing the similarity among models, e.g., the Euclidean distance. However, as indicated in [15,24,25], these distance-based methods show huge limitations for most non-convex models, especially for DNNs. The Kullback-Leibler divergence in [25] shows that it is able to measure the similarity between DNN models, but its asymmetry leads to more complex optimization. We choose the similarity measurement method based on the vector from the output layer to address these key points to compute the matching relationship between the current model and others. JS divergence is used to measure how one probability distribution is different from the other and is widely used in similarity measurement. The JS divergence of two DNN models (w_i, w_j) can be expressed as:

Algorithm 1: MGFL under T iterations

```
Input: Initial models of all the clients $w^0_{i\in\mathcal{N}\cup\mathcal{M}}$ with local iteration $t = 0$.
Output: Personalized models $w_{i\in\mathcal{N}\cup\mathcal{M}}$.
1  for $t$=1,2,...,$T$ do
2      for each client $i \in \mathcal{N} \cup \mathcal{M}$ do
3          Client $i$ solves local problem (2) and derive $w_i$. (Local model
           computation)
4      end
5      All the clients transmit their updated $w^t_{i\in\mathcal{N}\cup\mathcal{M}}$ and the local model
       accuracy $P_{i\in\mathcal{N}}$ to the server.
6      After server receiving $w^t_{i\in\mathcal{N}}$, $w_{i\in\mathcal{N}}$ can be obtained by
7      if in the early stage then
8          calculating (6). (Same-granularity FL)
9      else
10         if $t\%\theta == 0$ then
11             calculating (9). (Same-granularity FL)
12         else
13             calculating (9). (Cross-granularity FL)
14         end
15     end
16     Server calculates (6) after receiving $w^t_{i\in\mathcal{M}}$ and obtains $w_{i\in\mathcal{M}}$. (Server
       model aggregation)
17     The server broadcasts $w_{i\in\mathcal{N}\cup\mathcal{M}}$ to clients such that $w^t_i = w_i$.
18 end
```

$$\begin{gathered} JSD(w_i||w_j) = \frac{1}{2}P_i \log\frac{2P_i}{P_i + P_j} + \frac{1}{2}P_j \log\frac{2P_j}{P_i+P_j}, \\ P_i = g(w_i, x_i^S), \quad P_j = g(w_j, x_i^S), \end{gathered} \tag{3}$$

where x_i^S is the shared data of client i. $g(w_i, x_i^S)$ denotes the output of model w_i on the input data x_i^S. A smaller JS divergence indicates the more similar the two models are.

We then obtain a JS divergence matrix $D_{JS} \in \mathbb{R}^{N\times N}$, with $D_{JS}(i,j) = JSD(w_i||w_j)$. In order to construct the similarity matrix D, we first normalize each row of the JS divergence matrix and get the normalized matrix $\hat{D}_{JS}$, i.e.,

$$\hat{D}_{JS}(i,:) = \frac{(I - D_{JS}(i,:)) - \min\{I - D_{JS}(i,:)\}}{\max\{I - D_{JS}(i,:)\} - \min\{I - D_{JS}(i,:)\}}, \tag{4}$$

where I is a N-dimensional vector with all elements being 1. The similarity matrix are defined as the dot product of the normalized matrix and its transpose, which is shown as:

$$D = \sqrt{\hat{D}_{JS} \cdot \hat{D}^{\mathrm{T}}_{JS}}. \tag{5}$$

Although the JSD is symmetric in principle, the computed $\hat{D}_{JS}$ is not entirely symmetric, which is not conducive to reaching consensus on the client-side to make the aggregated model more generalized. Therefore, a practical approach is to use the inner product operation to create the ultimate similarity matrix completely symmetric. The server will aggregate the same-granularity models from clients according to the similarity matrix D to get new personalized models and return them to the clients for the next round of training. The aggregated results are shown below.

$$W = D \times W^t. \tag{6}$$

Similarly, the fine-granularity clients also perform same-granularity FL with execution details similar to those described above.

3.3 Cross-Granularity Federated Learning

Compared with the coarse-granularity model, the fine-granularity model is the higher level participant containing mature and complex knowledge, which results in the practical and logical cross direction of different granularity being fine to coarse. During the later stage of training, the server performs cross-granularity FL to facilitate clients' collaboration between different granularity, promoting more extraordinary performance for coarse-granularity models. We formulate the coarse-granularity model problem in the later stage as follows:

$$\begin{aligned} &\min_{W} \sum_{i=1}^{N} \frac{1}{N_i} \sum_{j=1}^{N_i} F_i(w_i, x_{i,j}, y_{i,j}) + \lambda \phi_i(w_i, x_{i,j}^S, y_{i,j}^S), \\ &\phi_i(w_i, x_{i,j}^S, y_{i,j}^S) = \sum_{r=1}^{|x_i^S|} \frac{\| \sigma(w_i, x_{i,r}^S) - \sigma(w_{h(i)}, x_{i,r}^S) \|^2}{| x_i^S |}, \end{aligned} \tag{7}$$

where $w_{h(i)}$ is the model of the fine-granularity client $h(i)$ that is selected to assist the coarse-granularity client i for its model performance improvement, $\sigma(\cdot)$ means the DNN models' output before the last dense layer. The second term $\phi_i(w_i, x_i^S, y_i^S)$ calculates the difference between the output vector before the coarse-granularity and the fine-granularity model's dense layer.

The premise of addressing the problem of Eq. (7) is to find out the correspondence between the coarse-granularity model and the fine-granularity model, i.e., how to ensure the value of $h(i)$ in Eq. (7). The optimal solution can be expressed as follows:

$$\begin{aligned} &h(i) = \mathop{\arg\max}_{m \in \mathcal{M},\ \epsilon(w_m, x_i^S, y_i^S) - A_i > 0} [\epsilon(w_m, x_i^S, y_i^S) - A_i], \\ &\epsilon(w_m, x_i^S, y_i^S) = \frac{\sum_{j=1}^{|x_i^S|} \mathbb{I}_{\{Hg(w_m, x_{i,j}^S) = y_{i,j}^S\}}}{|x_i^S|}, \quad m \in \mathcal{M}, \end{aligned} \tag{8}$$

where H is the prior knowledge matrix that contains the correspondence between coarse-granularity classes and fine-granularity classes. It is possible to transform fine-granularity classes into coarse-granularity classes using the matrix H.

$\epsilon(w_m, x_i^S, y_i^S)$ represents the conversion accuracy of the m-th fine-granularity model on the i-th coarse-granularity client's shared dataset. The function $\mathbb{I}(\cdot)$ is the indicator function and A_i is the accuracy of the coarse-granularity client i on the local dataset, which has been sent to the server.

To avoid redundant guidance, the coarse-granularity model only selects one fine-granularity model with the most prominent performance help for federated training at a time. The coarse-granularity model performs iterative weight updates in the server as:

$$w_i = w_i^t - \eta_s \nabla \phi_i(w_i, x_i^S, y_i^S), \tag{9}$$

where η_s is the learning rate in the server update process.

It is worthy to notice that the coarse-granularity model only updates parameters except for the dense layer and retrain the dense layer using shared data before returning it to the clients. The cross-granularity FL process only performs in later stage of the system, whose goal is to ensure that the fine-granularity models are mature enough to guide the coarse-granularity models.

4 Performance Evaluation

In this section, we evaluate the performance of MGFL and compare it with other methods. Our experimental evaluations focus on three aspects: performance on different data settings, improvement by cross-granularity FL, and the relationship between the same and cross-granularity models.

4.1 Dataset and Experiment Settings

We consider the CIFAR-100 dataset, which contains common things in life divided into 100 and 20 classes, corresponding to fine-granularity labels and coarse-granularity labels. Thus, each coarse-granularity class includes five fine-granularity classes. The dataset contains about 50000 training samples and 10000 testing samples uniformly distributed among coarse-granularity classes and fine-granularity classes.

In the evaluations, we set $N = 5$, $M = 5$ and number the clients under each granularity as 0–4. For coarse-granularity clients and fine-granularity clients with the same order (e.g., coarse-granularity client 0 and fine-granularity client 0), we set the same data distribution to facilitate the observation of relationship in cross-granularity FL. We adopt Wide-Resnet as the DNN model with multiple convolutional layers to serve for an image classification task. All evaluation results are implemented on Tensorflow 2.4.1 deployed on a server with Intel(R) Gold 6226R CPU, NVIDIA GeForce RTX 3090, and Windows Server 2019.

To compare the performance of MGFL and other methods on both IID and non-IID data, we apply different data settings. 1) IID data: each client holds the same distribution data; 2) non-IID data: each client holds samples of different 5–6 classes. It is assumed that clients hold the same distribution data belonging to a group, and the number of groups ranges from 2 to 5. Considering that tasks with different granularity in real life are usually deployed on different parties' clients, some parties that have similar data distribution while others have pretty different data distribution. That leads to the emergence of groups, so the data setting 2) is more common in practice. By default, when the confusion degree is 2, clients in {0, 1, 2} are a group with the same data distribution, and the other group consists of clients in {3, 4}. Similarly, the three groups are {0, 1}, {2, 3} and {4} at confusion degree 3, the four groups are {0, 1}, {2}, {3} and {4} at confusion degree 4, and each client is a group at confusion degree 5.

4.2 Experiment Results

Performance on Different Data Settings: We first evaluate the performance of MGFL under different data settings and compare it with other methods. Under the non-IID data, we set four confusion degrees *C2*, *C3*, *C4* and *C5* by dividing all clients into 2, 3, 4, and 5 groups. E.g., if all clients with the same granularity are divided into two groups, each contains 5–6 coarse-granularity classes or 25–30 fine-granularity classes. In that case, the confusion degree is two and is denoted by *C2*. We choose the best mean test accuracy (BMTA) [13] as the performance indicator defined as the highest mean test accuracy achieved overall communication rounds of training. Figure 3 shows the performance of coarse-granularity and fine-granularity models using four methods, MGFL, Alone, Cosine, and FedAvg [10], respectively, under training 300 epochs. The Cosine method replaces the JS divergence of MGFL with the cosine distance, and the Alone method only trains the model locally without FL.

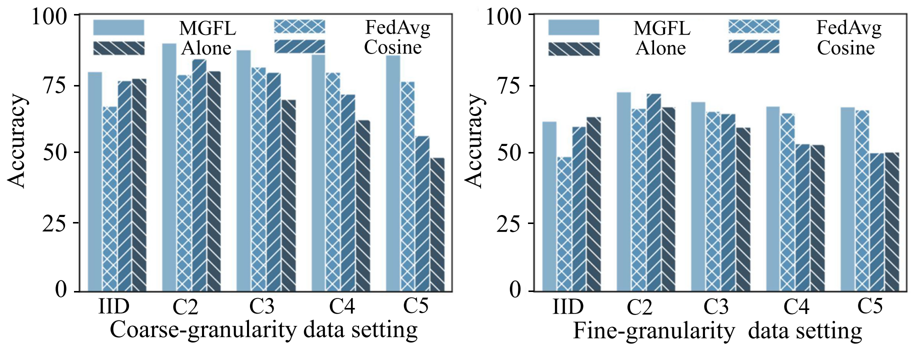

Fig. 3. BMTA of four methods under (a) coarse granularity and (b) fine granularity at different confusion degrees.

As shown in Fig. 3, MGFL exhibits superior performance on both coarse-granularity and fine-granularity clients, especially on the non-IID data. Since FedAvg was originally designed for clients with the same distribution, it is able to achieve high accuracy on IID data and performs poorly on non-IID data. MGFL shows high performance close to FedAvg on IID data and far better than it on non-IID data. Since the Cosine method uses the cosine distance commonly used in distance-based personalized FL [13,15], it captures the similarity among clients in low confusion degrees and has some advantages. However, it gradually loses its impact in the face of high confusion degrees. The Alone method shows the opposite trend to the Cosine method in that it performs the worst at the confusion degree 2. Moreover, it offers better performance as the confusion degree rises, indicating that the more complex the data, the more difficult it is to perform FL.

Performance Improvement by Cross-Granularity FL: Figure 4 shows the performance variation curves of coarse-granularity models in Group 1 and Group 2 at confusion degree 2. As shown in the figure, the clients' performance in both Group 1 and Group 2 using the MGFL method shows a significant increase and convergences to higher accuracy in the later stage than the other methods. The cross-granularity FL causes that in the later stage, where the fine-granularity model guides the coarse-granularity model resulting in a significant performance improvement.

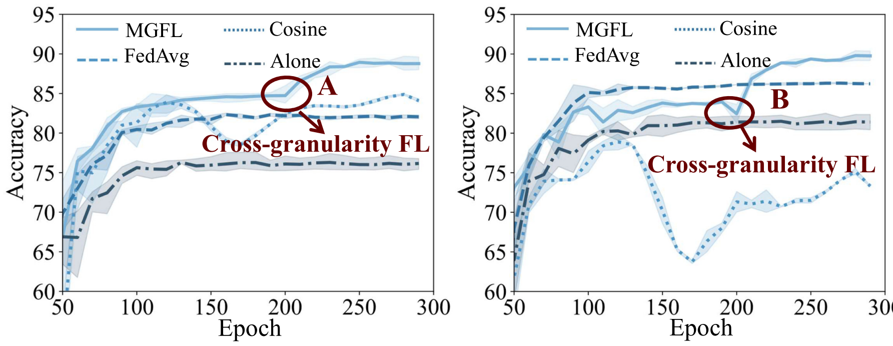

Fig. 4. The performance of coarse-granularity models in (a) Group 1 and (b) Group 2 at confusion degree 2.

It is worth noting that the performance of coarse-granularity models first degrades slightly at the beginning of the cross-granularity FL, as shown in Fig. 4 points A and B, due to improper coordination and incompatibility. That also shows that the parameters of the fine-granularity model are not fully applicable to the coarse-granularity model, so the parameters of the coarse-granularity model do not be directly replaced by those of the fine-granularity model. Even parameter approximation (shown in Fig. 1) requires many rounds of iterations to eliminate the incompatibility and improve performance. In the later stage, fine-granularity models perform the same-granularity FL as in the early stage except for guiding coarse-granularity models. As shown in Fig. 5, the overall performance of MGFL is better than other methods and has minor performance fluctuation in both Group 1 and Group 2.

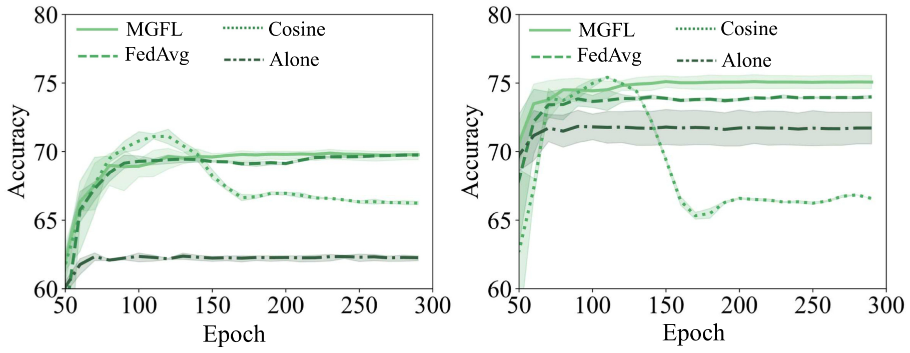

Fig. 5. The performance of fine-granularity models in (a) Group 1 and (b) Group 2 at confusion degree 2.

Relationship in the Cross-Granularity FL: Selecting the most relevant fine-granularity model for the coarse-granularity model is a critical step in the cross-granularity FL. Figure 6 shows the selection of each coarse-granularity client to fine-granularity client in each execution of cross-granularity FL at confusion degrees 2, 3, 4.

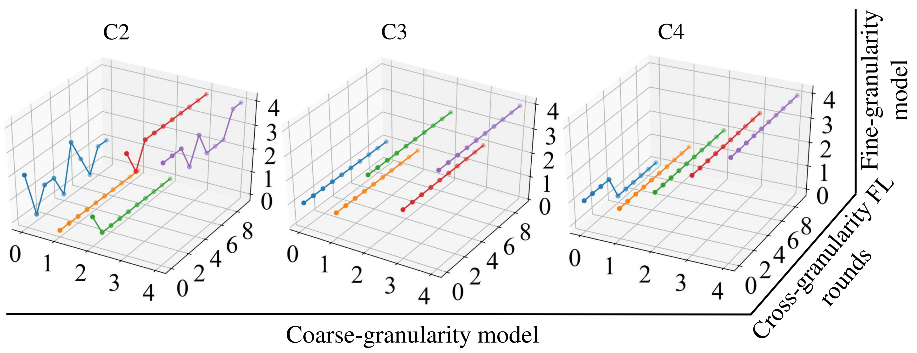

Fig. 6. Relationships in cross-granularity FL at different confusion degrees.

As shown in the figure, MGFL can precisely choose the most similar fine-granularity model to guide fine-granularity one in each cross-granularity FL. E.g., coarse-granularity client 0 at $C2$ selects the clients in group {0, 1, 2} due to keeping the same data distribution. Though selection becomes more complex as the number of groups increases, MGFL has the ability to overcome the complexity and make the right choices. Another observation based on the results is that the smaller the confusion degree, the more fluctuating the clients' selection when performing the cross-granularity FL. Compared to the results at confusion degree 4, the coarse-granularity clients at confusion degree 2 fluctuate considerably with different clients selected in each cross-granularity FL. The more the number of groups, the fewer the number of clients in the group and the fewer the number of clients to select from, resulting in more minor fluctuations in selection.

Relationship in the Same-Granularity FL: We aim to leverage the inherent group relationship learned by MGFL to select similar models for aggregation in the same-granularity FL while improving the overall accuracy performance. We visualized the similarity matrix of MGFL and Cosine methods in the early rounds at different confusion degrees, exhibiting the group structure. As shown in Fig. 7, compared with the Cosine method, the group structure in our framework is generally learned precisely in the early rounds, which can be interpreted as having closer similar values within the same group. E.g., the nine values in the upper left corner of MGFL at *C2* are superior to the nine values in the upper left corner of Cosine at *C2*. Besides, our method captures the same-granularity relationship more accurately, especially as the confusion degree increases.

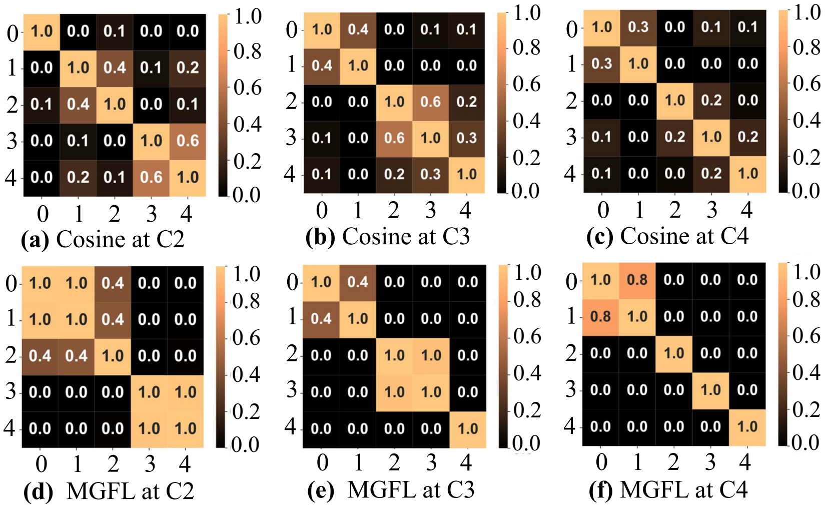

Fig. 7. Relationships in same-granularity FL using (a)–(c) the Cosine and (d)–(f) the MGFL methods at different confusion degrees.

Although learning group relationships become more difficult as the degree of confusion increases, MGFL can overcome this difficulty and learn relationships more accurately than the Cosine method. Due to the limitation of the length of the paper, we only show the results of the coarse-granularity models, and the behaviors of the fine-granularity models are similar to that of the coarse-granularity model.

5 Conclusion

In this paper, we address the challenge of collaborative training among multi-granularity clients. Specifically, we develop a novel method named MGFL that introduces two mechanisms for fine-granularity clients and coarse-granularity clients. MGFL designs personalized models for different-granularity clients using a divergence-based similarity method in same-granularity FL and breaks the restriction cross-granularity FL by adjusting the empirical risk loss function. Extensive experiments show that our method out-performs other baselines on both IID and Non-IID data and proves its superior performance even under confusing data distribution.

Acknowledgement. This work is supported by research on key technologies of electrical cloud-edge-end collaborative AI model sharing in Science and Technology Project of State Grid Headquarters (2021, Power base support technology - 30).

References

1. Wang, T., Luo, H., Zeng, X., Yu, Z., Liu, A., Sangaiah, A.K.: Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities. IEEE Trans. Intell. Transp. Syst. **22**(3), 1797–1806 (2020)
2. Wang, T., et al.: Propagation modeling and defending of a mobile sensor worm in wireless sensor and actuator networks. Sensors **17**(1), 139 (2017)
3. Wang, X., Li, R., Wang, C., Li, X., Taleb, T., Leung, V.C.: Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. IEEE J. Sel. Areas Commun. **39**(1), 154–169 (2020)
4. Wang, X., Wang, C., Li, X., Leung, V.C., Taleb, T.: Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching. IEEE Internet Things J. **7**(10), 9441–9455 (2020)
5. Wang, X., Han, Y., Leung, V.C., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surv. Tut. **22**(2), 869–904 (2020)
6. Alsheikh, M.A., Niyato, D., Lin, S., Tan, H., Han, Z.: Mobile big data analytics using deep learning and apache spark. IEEE Netw. **30**(3), 22–29 (2016)
7. Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G., Rogers, R.: Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984 (2018)
8. Bonawitz, K., et al.: Practical secure aggregation for federated learning on user-held data. arXiv preprint arXiv:1611.04482 (2016)
9. Wang, T., Bhuiyan, M.Z.A., Wang, G., Qi, L., Wu, J., Hayajneh, T.: Preserving balance between privacy and data integrity in edge-assisted internet of things. IEEE Internet Things J. **7**(4), 2679–2689 (2019)
10. Hard, A., et al.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)
11. Kairouz, P., et al.: Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019)
12. Tan, A.Z., Yu, H., Cui, L., Yang, Q.: Towards personalized federated learning. arXiv preprint arXiv:2103.00710 (2021)

13. Huang, Y., et al.: Personalized cross-silo federated learning on non-IID data. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 7865–7873 (2021)
14. Nagalapatti, L., Narayanam, R.: Game of gradients: mitigating irrelevant clients in federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9046–9054 (2021)
15. Sattler, F., Müller, K.R., Samek, W.: Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Trans. Neural Netw. Learn. Syst. **32**, 3710–3722 (2020)
16. Ouyang, X., Xie, Z., Zhou, J., Huang, J., Xing, G.: ClusterFL: a similarity-aware federated learning system for human activity recognition. In: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, pp. 54–66 (2021)
17. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: 2019 IEEE International Conference on Communications (ICC), ICC 2019, pp. 1–7 (2019)
18. Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.: Federated multi-task learning. arXiv preprint arXiv:1705.10467 (2017)
19. Liu, F., Wu, X., Ge, S., Fan, W., Zou, Y.: Federated learning for vision-and-language grounding problems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11572–11579 (2020)
20. Corinzia, L., Beuret, A., Buhmann, J.M.: Variational federated multi-task learning. arXiv preprint arXiv:1906.06268 (2019)
21. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018)
24. Ghosh, A., Hong, J., Yin, D., Ramchandran, K.: Robust federated learning in a heterogeneous environment. arXiv preprint arXiv:1906.06629 (2019)
25. Jiang, B., Pei, J., Tao, Y., Lin, X.: Clustering uncertain data based on probability distribution similarity. IEEE Trans. Knowl. Data Eng. **25**(4), 751–763 (2011)