# Towards Fairer and More Efficient Federated Learning via Multidimensional Personalized Edge Models

Yingchun Wang School of Comp. Sci.&Tech. Xi'an Jiaotong University Xi'an, Shaanxi, China 20116342r@connect.polyu.hk

Song Guo Department of Computing The Hong Kong Polytechnic University Hong Kong SAR, China song.guo@polyu.edu.hk Jingcai Guo Department of Computing The Hong Kong Polytechnic University Hong Kong SAR, China jc-jingcai.guo@polyu.edu.hk

> Weizhan Zhang School of Comp. Sci.&Tech. Xi'an Jiaotong University Xi'an, Shaanxi, China zhangwzh@xjtu.edu.cn

Jie Zhang Department of Computing The Hong Kong Polytechnic University Hong Kong SAR, China 18104473r@connect.polyu.hk

Qinghua Zheng School of Comp. Sci.&Tech. Xi'an Jiaotong University Xi'an, Shaanxi, China qhzheng@mail.xjtu.edu.cn

Abstract—Federated learning (FL) is an emerging technique that trains massive and geographically distributed edge data while maintaining privacy. However, FL has inherent challenges in terms of fairness and computational efficiency due to the rising heterogeneity of edges, and thus usually results in sub-optimal performance in recent state-of-the-art (SOTA) solutions. In this paper, we propose a Customized Federated Learning (CFL) system to eliminate FL heterogeneity from multiple dimensions. Specifically, CFL tailors personalized models from the specially designed global model for each client jointly guided by an online trained model-search helper and a novel aggregation algorithm. Extensive experiments demonstrate that CFL has full-stack advantages for both FL training and edge reasoning and significantly improves the SOTA performance w.r.t. model accuracy (up to 7.2% in the non-heterogeneous environment and up to 21.8% in the heterogeneous environment), efficiency, and FL fairness.

*Index Terms*—Federated Learning, Edge Computing, Neural Architecture Search, Model Compression, Deep Learning.

#### I. INTRODUCTION

Machine deep learning has made tremendous success in the past few years across multiple real-world applications [1]–[34]. In recent years, with the advances in the computing capability of edge devices, more and more machine learning models are usually deployed locally to directly perform training or inference. At the same time, the explosive growth of end-user data can have a high potential to bring about tangible benefits for various applications, i.e., user verification [35], [36], self-driving [37], human activity recognition [38], medical health monitoring [39], and so on. However, the data resources are usually geographically distributed across different edge devices, this is then challenging to train a deep model collaboratively on both privacy preservation and transmission overhead reduction. Worse still, the "isolated data islands" could be seriously heterogeneous in terms of data distribution,

quality, and quantity. Therefore, how to efficiently excavate the rich knowledge hidden behind these distributed data and train an accurate cooperative model in a privacy-protection way has become a crucial research topic.

To address the above challenges, federated learning (FL) has emerged as a promising paradigm of privacy-preserving distributed machine learning framework [40], [41]. FL enables edge devices to train a model locally based on its own dataset and communicates with other models in the server for aggregating a more generalized global model, without collecting and sharing any privacy-sensitive information from users. However, existing FL methods mostly use a unified global model for all participants and ignore the potential hardware and data heterogeneities between them, i.e., diverse hardware specifications, different network conditions, highly biased data, and inconsistent data qualities.

In recent years, there have been some works focusing on eliminating the FL heterogeneity and obtained some promising results [42]-[45]. However, most of them suffer from sub-optimization in both model performance and training efficiency, for only addressing the above problems in a single-dimensional perspective, i.e., data heterogeneity. Such a scheme may have three limitations: 1) the computation and transmission stragglers among different clients usually lead to extremely low training efficiency; 2) the communication overhead of exchanging the updates of full models may cause excessive transmission delays, especially for large deep neural networks; and 3) the heterogeneities can lead to biased training across clients and introduce significant performance unfairness. In this paper, we propose a novel Multidimensional Customized Federated Learning (CFL) system to achieve a fairer and more efficient FL. The main idea of CFL is to strengthen the identity between the FL participants by minimizing the impact of multi-dimensional heterogeneities on FL. Specifically, a specially designed data-quality aware model with a tiny reinforcement learning (RL) module is regarded as the common model in CFL. Then, CFL tailors a personalized model from the common one for each client according to the corresponding device's hardware specification and data condition. The model tailoring is achieved by an online trained model-search helper which can output the submodel, i.e., network sub-structure, with the best accuracy and less latency for the current client. Moreover, to better aggregate information from asymmetric personalized models, we propose a novel parameter aggregation algorithm via module scaling and alignment in each FL round. Therefore, the personalized models can be sampled dynamically from the common model with the data-aware RL module, and be trained further in an FL paradigm with the new scaling aggregation method. Extensive experiments on CIFAR-10 and MNIST datasets (processed as mixed quality) demonstrate that the proposed CFL can achieve better task accuracy, fairer model performance, and higher training acceleration against representative FL methods.

- We propose a new FL paradigm that tailors customized submodels for heterogeneous edge workers, which obviously reduces the time differences of local training on different devices and greatly speeds up the federated training stage.
- We design a novel search helper to select customized models for different workers and a new model aggregation algorithm to aggregate updates with different architectures. Both proposals enable the use of FL in essentially heterogeneous edge computing.
- We improve the parent model to be data quality-aware by adding an RL module and training it on special process datasets with different data qualities. It turns out that data quality-aware FL models perform better on real-world applications in edge computing scenarios than traditional FL paradigms.

#### II. RELATED WORKS

## A. Model Compression

Model compression has been extensively studied to reduce the computation overhead and memory usage, so that neural models can be better deployed on resource-limited edge devices [46], [47]. It is essential to do a trade-off between the compressed rate and the accuracy reduction. Current popular methods include pruning [48] [49] [50], quantization [51], parameter sharing [52], knowledge distillation, low-rank approximation and direct design of compact models, etc. We compare our work with several mainstream-related works including standard FL [40], MT-FL [53], and Model compression. The details are shown in Table I.

## B. Federated Learning

Applying big data analysis and artificial intelligence (AI) in practical scenarios is not easy since high-quality datasets are rare or difficult to access. To deal with the data island problem and utilize the distributed data from different mobile devices, Google has proposed federated learning (FL) [40], [54] in 2016 to train machine learning models over decentralized devices. Such a paradigm aims to mine distributed datasets without sharing privacy-sensitive data, and thus is deemed to greatly help launch AI in more areas. FL requires a set of workers to cooperate with the coordinator server, which assigns the model parameters to each worker to train the local model and collects the updated parameters to accumulate to the shared model repeatedly. Current focuses of FL research include user privacy preservation [55], incentive scheme to promote collaboration between multiple users [56], communication overhead [57] and fairness optimization [58].

## C. Heterogeneity in Data and Devices

Data heterogeneity here is defined as "the datasets from different devices are often statistically deficient (non-IID), e.g., of different label distribution, dataset size and sample noise level, etc". Device heterogeneity refers to the fact that different devices participating in FL tasks are highly likely to possess quite different levels of hardware specification and environmental conditions, e.g., different CPU spec, memory size, and network bandwidth. Heterogeneity in distributed scenarios leads to inevitable waiting latency for parameter aggregation as well as the variance of model accuracy among different devices. Lots of works have been proposed to deal with such problems. For example, the vertical FL is proposed to deal with feature heterogeneity [59].

Several approaches to personalized federated learning [60]– [62] have been proposed to address the heterogeneity problem in FL. For example, Zhao et al. [63] used the method of data enhancement to reduce the statistical heterogeneity between customer data sets and strengthen the training of the whole play model by sharing some balanced global data among drifting clients. Li et al. [64] proposed the FedMD method to train a global model with the assistance of a public dataset and allow each client to fine-tune with its own private dataset. However, most of the above methods only consider the heterogeneity of data distribution between clients during training for personalized design, ignoring the common heterogeneous problems such as hardware and data quality. How to incorporate diverse heterogeneities remains a serious challenge in FL.

#### III. METHODOLOGY

# A. An Overview of CFL

The pipeline of CFL is illustrated in Figure. 1. Specifically, CFL can be divided into the interwoven local training and global updating process that follows a three-part paradigm: **submodels sampling** (on the server), **local training** (on the client), and **model updating** (on the server).

First, in the submodels sampling stage, the server samples a personalized network sub-structure for each FL worker according to its hardware specification and data quality, wherein the hardware specification is defined by the device model, and the data quality is quantized to five different quality levels according to the Gaussian Blur. To be specific, the



Fig. 1. The overview of the CFL. The server part consists of three main components, i.e., the online accuracy predictor, the latency lookup table, and the expansion module, which work jointly to sample the optimal submodel for each client and update the global model. The blue lines on the parent model represent additional RL gating modules. Sam., Avg., and Ref. denote submodel sampling, parameter averaging, and the referenced module updating process respectively.

| Methods     |        | leterogeneous in training | Heterogeneous in inference                  |        |      |                                     |  |
|-------------|--------|---------------------------|---|--------|------|-------------------------------------|--|
|             | Device | Data                      | Idea  | Device | Data | Idea                                |  |
| Standard FL | N      | N                         | All clients train the same model            | N      | N    | All clients use the same            |  |
| [40]        |        |                           |   |        |      | trained model                       |  |
| MT-FL [53]  | Ν      | v                         | Same structure for collaboration            | N      | Y    | Each client uses its specific model |  |
|             |        | 1                         | Different parameters for data heterogeneity |        |      |                                     |  |
| Model       | N      | N                         | Training a large model in cloud             | Y      | N    | Compress the large model into       |  |
| compression |        |                           |   |        |      | a small model                       |  |
| [52]        |        |                           |   |        |      |                                     |  |
| CFL (ours)  | Y      | Y                         | Different structures for both types of      | Y      | Y    | Each client uses its specific model |  |
|             |        |                           | heterogeneity                               |        |      |                                     |  |
|             |        |                           | The same parameters for collaboration       | ]      |      |                                     |  |

 TABLE I

 Comparison of some works related to edge computing

submodels are first sampled from the global (parent) model using "Genetic Algorithm" on both model depth and width dimensions. After that, the selected submodel will be further filtered through a search helper composed of an online-trained accuracy predictor and an offline latency lookup table [65].

Next, the sampled submodels are assigned to different clients and be trained with heterogeneous local datasets separately. At the last epoch of local training, the test accuracy of each submodel is collected as the training profile, which is then be uploaded along with the model gradients.

Last, there are two model updating processes parallelly conducted in the server. The first update is about the **global model**, which aggregates all the uploaded local updates from clients by the specially designed aggregation algorithm. The second update is about the **accuracy predictor**, which judges the accuracy that the current submodel can achieve based on the hardware and data conditions reported by each client.

#### B. Personalized Models For Device Heterogeneity

In this subsection, we mainly focus on problems introduced by device heterogeneity in traditional federated learning during the training stage. Specifically, there are two main problems. The first comes from the selection of customized submodels, one of the filters of which needs an accurate and less training overhead accuracy predictor. And another challenge would be the above-mentioned submodels aggregation. These two issues are respectively discussed in the following two subsections below.

1) submodels Sampling: The submodel sampling process is based on two sequential steps, including submodel searching and filtering. To be specific, submodels are firstly randomly generated using genetic algorithms in a two-dimensionallimited search space. The details can be found in Algorithm 1. After that, these generating results will be further filtered through a search helper composed of an online-trained accuracy predictor and an offline latency lookup table [65], wherein, the accuracy predictor is essentially a four-layer linear classifier and is dynamically trained in the first several FL rounds. Its training datasets are the above-mentioned training profiles, which are formed by using the submodel structure information and data quality as the data sample, and the test accuracy as the data label. Since the accuracy predictor is a simple linear classifier and the number of clients in FL is large, the data samples and labels collected from one or two CFL

# Algorithm 1 submodels Selection

**Input:** Accuracy predictor  $f_t$ , computational latency table g, parent model  $\omega_t$ , search times S, computational latency bound  $l_k$ , hardware profile  $p_k$ , data quality  $q_k$ , number of workers K**Output:** Customized models  $\omega_k^t$  for each worker kInitialize  $acc_k = 0, \forall k = 1, ..., K$ ; for i = 0, ..., S do for k = 0, ..., K do

Select a submodel  $\omega$  from  $\omega_t$  with the bounded latency on worker k as  $g(\omega, p_k) < l_k$ ; if  $f_t(\omega, q_k) > acc_k$  then  $acc_k = f_t(\omega, q_k)$ ;  $\omega_k^t = \omega$ ; end if end for

end for

| Algorithm 2 Training Accuracy Predictor                     |
|---|
| <b>Input:</b> The accuracy predictor <i>f</i>               |
| <b>Output:</b> $f$ in each FL round                         |
| for $t = 1$ to $T$ do                                       |
| for $k = 1$ to K do   |
| Collect $q_k$ , $\omega_k^t$ , $acc_k^t$ from the worker k; |
| Construct sample $x_k = (q_k, \omega_k^t), y_k = acc_k^t$ ; |
| end for   |
| Use all K samples $(x, y)$ to update the accuracy predictor |
| $f_t$ for one epoch;  |
| end for   |

rounds would be sufficient enough to let the accuracy predictor converge or reach a satisfying prediction accuracy. Once the training of the accuracy predictor starts to converge, or its test accuracy reaches a predefined threshold, its training can be stopped to stabilize submodels as well as reduce overhead.

In subsequent rounds, the predicted accuracy, together with the latency table, are used for the selection process. A complete training process in one CFL round of the accuracy predictor is shown in Algorithm 3.

2) submodels Aggregation: To solve the challenges of the aggregation of structurally misaligned personalized submodels, we proposed a novel aggregation algorithm that expands and aligns all submodels before the actual aggregation. Same with model sampling, the expansion of submodels is also limited in depth and width dimension. The details are as follows.

• Layer group: Since a residual model architecture is used in the parent model, different parameter settings and activation functions are used in different residual blocks. Before the actual model depth and layer width alignment, a layer grouping process is necessary to maintain the same parameter distribution as the parent model before and after expanding operation. Specifically, all the convolution layers except the first one of the submodels are divided into different groups according to the residual



Fig. 2. Depth expansion of submodel.



Fig. 3. Width expansion of customized models. The left rectangle represents the width of layers in the global model. The right rectangle represents the width of the layers in the submodels.

settings of the server model to achieve a group-level alignment.

- Width expansion: Since the channels of each layer of the submodels are randomly selected from the parent model according to the limitations of the accuracy predictor and latency lookup table. In the vanilla FL aggregation stage, it is the parameters of the channels at the same location in different submodels participate in one operation. To face the challenge of the disordered channels of various submodels that are scrambled during the sampling process, as shown in Fig. 3, all channels in the submodels are first sorted in the original order to keep the consistency of the parent model structure. And then, after the sorting process, an expansion operation is performed for each layer of the submodels, if its current layer width is smaller than the width of the large model, all 0 channels will be added to fill the current layer to its original width in the parent model. Thus far, all submodels have achieved width alignment.
- Depth expansion: Depth alignment is performed groupwise and can be done only after the group alignment. As shown in Fig. 2, for those groups with fewer layers in submodels than the parent model, they are padded with all 0 layers to reach the layer number of the parent standard. Furthermore, the width and kernel size in these all-zero layers are the same as the width of the corresponding layer of the parent model.

After the model alignment, the federated average algorithm could be performed to aggregate all of the uploaded local grads and update the global model. The detailed working flow of the whole alignment and aggregation operation is given in Algorithm 3.

| Algorithm 3 submodel Alignment and Aggregation                             |
|--|
| <b>Input:</b> The number of workers K, submodel update $\Delta_k^t$ and    |
| data size $n_k$ for each worker k, and total data size n;                  |
| <b>Output:</b> Global model $\omega_t$ ;                                   |
| for $k = 1$ to K in parallel do  |
| Group the layers of the update $\Delta_k^t$ by block;                      |
| Expand the width of the layers of $\Delta_k^t$ ;                           |
| Extend the depth of $\Delta_k^t$ ;   |
| end for  |
| Aggregate all updates $\Delta_t = \sum_{k=1}^K \frac{n_k}{n} \Delta_k^t$ ; |
|  |

# C. RL-Based parent model For Data Heterogeneity

After the training stage of FL, the optimized models (or submodels) would be deployed on edge devices in the real world. And a machine learning model in the wild (e.g., a self-driving car) must be prepared to make sense of its surroundings in rare conditions that may not have been well-represented in its training set. However, the previous personalized FL only focuses on the heterogeneity of data distribution and ignores the changes in data quality in practice.

Previous work [66] has proved that to achieve the same prediction accuracy, heterogeneous data quality requires different network complexity. For example, a clear image may only require a smaller neural network to achieve the same accuracy as a blurred image.

Thus, to achieve fairer task performance across heterogeneous datasets, we enable the global model to be data-aware by an RL gating module to assign personalized submodels according to different data conditions. The RL function could dynamically select which layers of a convolutional neural network should be skipped during submodel sampling. Specifically, we first introduce layer-wise RL agents which are coded as a function from the feature activations to the probability distribution over the skipping action. Note that to cope with non-differentiable data-aware model sampling decisions, we first warm up the global model and train it using a hybrid learning algorithm combining supervised learning and reinforcement learning [66]. It turns out that the RL modules in CFL not only speed up the FL edge training but also could accelerate the following inference stage by assigning more lightweight models to edge devices. In a word, CFL is a great full-stack FL system that could greatly reduce the computing overhead in both the training and inference stages.

Last, the overall process of our method is summarized in Algorithm 4.

#### IV. EXPERIMENT

#### A. Benchmark

**Dataset**: We use CIFAR-10 and MNIST datasets as the baseline, and a set of related datasets are extended from these raw datasets to emulate two different data heterogeneities, such as **data quality heterogeneity** and **data distribution** 

Algorithm 4 CFL: Customized Architecture Search based Federated Learning

| <b>Input:</b> Number of workers $K$ , learning rate $\eta$                             |
|--|
| <b>Output:</b> local updates $\Delta_k^t$ and test accuracy of worker k in             |
| communication round t  |
| <b>Initialize:</b> parent model $\omega_1$ ;   |
| for $t = 1$ to $T$ do  |
| On server:   |
| Select the submodel $\omega_k^t$ from $\omega_t$ for each worker k by                  |
| using the search helper;   |
| Send submodels $\omega_k^t$ to all workers ;   |
| Receive and aggregate $\Delta_k^t$ of all workers to get $\Delta_t$ ;                  |
| Update the global model $\omega_{t+1} = \omega_t - \Delta_t$ ;                         |
| Receive data and hardware profile and use them to update                               |
| the search helper;   |
| <b>On worker</b> k=1,,K:   |
| Receive submodel $\omega_k^t$ from server;   |
| for epoch $e$ ranges from 1 to $E$ do  |
| Compute the stochastic gradient $\nabla l(\omega_{k,e-1}^t)$ from a                    |
| random mini-batch;   |
| Update submodel $\omega_{k,e}^t = \omega_{k,e-1}^t - \eta \nabla l(\omega_{k,e-1}^t);$ |
| end for  |
| Compute local update $\Delta_k^t = \omega_{k,E}^t - \omega_{k,0}^t$                    |
| Send $\Delta_k^t$ to server;   |
| Send test accuracy and hardware specification profile to                               |
| server;  |
| end for  |

heterogeneity. For quality heterogeneity, the raw dataset is independently and identically (IID) divided into several batches and processed by Gaussian blurring and image sharpening with different levels. These batches are re-mixed to form new mixed-quality datasets. For distribution heterogeneity, each dataset is randomly divided into 32 Non-IID subsets. The data class imbalance degree is set to 0.8 in this work, i.e., 80% of each worker's local data belongs to the same class, and the remaining 20% are evenly selected from the remaining categories. CIFAR-10 dataset is processed to simulate the data quality heterogeneity. It consists of 60000 32x32 color images in 10 classes, with 6000 images per class and 10000 images per batch. There are five batches of images for training and one batch of images for the test. To simulate data quality heterogeneity during practical inference, we use three different degrees of Gaussian blur and image sharpening on the CIFAR-10 dataset, one per batch, to produce datasets of different quality but the same distribution. To be specific, we divided the training set of CIFAR-10 into five groups: unprocessed, three degrees of Gaussian blur, and sharpening images respectively. Instead of using complex data quality metrics, we apply different variances of the added noise to represent the heterogeneity in data quality. MNIST dataset is used to generate data heterogeneity in both distribution and quality. As for the data quality heterogeneity, we divide the MNIST training set uniformly into five IID subsets and conduct the abovementioned Gaussian blur or image sharpening per group, to

produce datasets of different quality but the same distribution. As for the data distribution heterogeneity, we divide the whole MNIST dataset randomly into 32 Non-IID subsets to simulate the different data distribution between federated workers. The data class imbalance degree is set to 0.8 in this work, i.e., 80% of each worker's local data belongs to the same class, and the remaining 20% are evenly selected from the remaining categories. For the parent model, it is pre-trained on quality heterogeneous IID datasets, and then federally trained on quality heterogeneous and Non-IID datasets.

**Model**: We use a once-for-all network [65] with layer-wise RL gate as the parent model, which is built on MobileNetV3 with elastic depth, width, and input size. All of the customized submodels in our experiments will be selected from the parent model.

## B. The Comparison of CFL with FL SOTA

In this section, we compare the performance of CFL using personalized models and FL SOTAs using one global model (abbreviated as FL in the following) on two different data heterogeneity settings with respect to data quality and distribution. The results are shown in Figure. 4 (a) and Figure.4 (b), respectively. It is obvious that CFL performs significantly better than FL under both heterogeneous settings, especially when the data quality is heterogeneous. Figure. 5 shows the time required for the first 200 iterations over 32 clients of CFL and FL, respectively.

The results demonstrate that CFL not only significantly improves training efficiency but also improves FL fairness because the training time difference between clients is significantly reduced.



(b) Distribution heterogeneity

Fig. 4. (a) Quality heterogeneity. (b) Distribution heterogeneity.

# C. Federated Learning vs. Independent Learning

The performance gain from CFL to local independent learning (IL) using customized models is demonstrated. The training experiments are conducted over two categories of



Fig. 5. Time required for 200 iterations on 32 workers.



(b) Distribution heterogeneity

Fig. 6. Comparison between CFL and independent learning with customized submodels. (a) Quality Heterogeneity: MNIST and 32 workers. (b) Distribution Heterogeneity: MNIST and 32 workers.

datasets, i.e., MNIST with both heterogeneous quality and heterogeneous distribution, which are split into 32 subsets for each worker respectively. The result of the accuracy comparison between CFL and independent local training is shown in Table II, it is obvious that using CFL in model training for edge computing consistently outperforms independent training in both heterogeneous and non-heterogeneous data distributions. And, in the case of data heterogeneity, the advantages of CFL are further amplified.

TABLE II COMPARISON OF TEST ACCURACY UNDER TWO EDGE COMPUTING SETTINGS: CFL AND INDEPENDENT LOCAL TRAINING.

|          | Non-heterog | geneous Data | Heterogeneous Data |        |  |
|----------|-------------|--------------|--------------------|--------|--|
| scenario | CFL (%)     | IL (%)       | CFL (%)            | IL (%) |  |
| worker 0 | 82          | 74.8         | 80.5               | 68.7   |  |
| worker 1 | 73.6        | 73.4         | 72.6               | 50.8   |  |
| worker 2 | 86.1        | 82.3         | 85.6               | 69     |  |

(i) *Heterogeneous Data Quality:* We generate Gaussian fuzzy data with three fuzzy degrees, unprocessed data, and sharpening data for all workers (randomly assigned), and then conduct the CFL and independent learning respectively. The



(c) Worker 2 with sharpening data



(d) Computation Percentage

Fig. 7. Comparison of accuracy and computational cost between the FL model using data quality-aware (with RL gate) and the common FL model without awareness under different data quality.

results are shown in Fig 6(a). It is verified that the final test accuracy of the customized models in CFL is obviously higher than that of the independent learning method. This is because in CFL workers can learn from the experiences of others, which is absent in independent training. In reality, different edge devices often have heterogeneous data.

(ii) *Heterogeneous Data Distribution:* Fig 6 (b) shows that the final test accuracy of CFL is higher than that of independent learning. Because in CFL, workers can learn and improve the local model from the parameter aggregation operation, while it is not possible for independent training.

## D. Data quality-aware Parent Model

In this section, we show how the RL gates can benefit the parent model. We set different data qualities for different workers. After deploying RL gates on each layer of the original parent model, we first train it in advance on the server using a small public dataset with uniformly distributed categories and the worst data quality. This pre-trained model can then be used for submodel selection initially. The test accuracy of customized models is shown in Fig. 7(a-c). The results show that the RL gate-enabled submodel selection not only consumes less time to converge but also reduces both the training and inference time since the computation of some layers of customized models is waived. To better demonstrate the computation overhead reduction, the computational percentage curves in the training phase of the customized models for three workers are given in Fig. 7(d). This percentage is defined as the ratio of the number of layers that are actually calculated to the number of all layers of the model.

#### V. CONCLUSION

In this paper, we introduce a novel customized federated learning framework, which first takes the multi-dimensional heterogeneity in federated learning. Specifically, we design a novel aggregation algorithm to reduce the calculation delay and accuracy difference between the cooperative FL devices. What's more, CFL uses a specially designed data qualityaware central model via RL gate to accelerate reasoning and improve robustness in the face of data quality changes. Extensive experiments have proved the effectiveness of CFL.

#### REFERENCES

- J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, Ieee, 2009.
- [2] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," ACM SIGKDD explorations newsletter, vol. 6, no. 1, pp. 20–29, 2004.
- [3] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, "Machine learning in financial crisis prediction: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 421– 436, 2011.
- [4] J. Guo and S. Guo, "A novel perspective to zero-shot learning: Towards an alignment of manifold structures via semantic feature expansion," *IEEE Transactions on Multimedia*, vol. 23, pp. 524–537, 2020.
- [5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions* on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2014.
- [6] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers), pp. 1354–1364, 2015.
- [7] J. Guo, S. Ma, J. Zhang, Q. Zhou, and S. Guo, "Dual-view attention networks for single image super-resolution," in *Proceedings of the 28th* ACM International Conference on Multimedia, pp. 2728–2736, 2020.
- [8] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xdeepfm: Combining explicit and implicit feature interactions for recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1754–1763, 2018.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision* and pattern recognition, pp. 770–778, 2016.
- [10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
- [11] J. Guo and S. Guo, "Adaptive adjustment with semantic feature space for zero-shot recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3287–3291, IEEE, 2019.
- [12] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proceedings of the IEEE international conference on computer vision*, pp. 3676–3684, 2015.

- [13] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
- [14] D. Mahapatra, B. Bozorgtabar, and Z. Ge, "Medical image classification using generalized zero shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3344–3353, 2021.
- [15] J. Guo, S. Guo, Q. Zhou, Z. Liu, X. Lu, and F. Huo, "Graph knows unknowns: Reformulate zero-shot learning as sample-level graph recognition," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI-23*, vol. 2, 2023.
- [16] Z. Liu, S. Guo, J. Guo, Y. Xu, and F. Huo, "Towards unbiased multilabel zero-shot learning with pyramid and semantic attention," *IEEE Transactions on Multimedia*, 2022.
- [17] J. Guo, S. Guo, S. Ma, Y. Sun, and Y. Xu, "Conservative novelty synthesizing network for malware recognition in an open-set scenario," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [18] X. Lu, Z. Liu, S. Guo, and J. Guo, "Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [19] Y. Wang, J. Guo, S. Guo, and W. Zhang, "Data quality-aware mixedprecision quantization via hybrid reinforcement learning," *arXiv preprint arXiv:2302.04453*, 2023.
- [20] Z. Liu, S. Guo, X. Lu, J. Guo, J. Zhang, Y. Zeng, and F. Huo, "(ml)2pencoder: On exploration of channel-class correlation for multi-label zero-shot learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [21] J. Guo and Z. Liu, "Application: Image-based visual perception," in Machine Learning on Commodity Tiny Devices, pp. 123–144, CRC Press, 2023.
- [22] F. Huo, W. Xu, J. Guo, H. Wang, Y. Fan, and S. Guo, "Offline-online class-incremental continual learning via dual-prototype self-augment and refinement," arXiv preprint arXiv:2303.10891, 2023.
- [23] Q. Zhou, Z. Qu, S. Guo, B. Luo, J. Guo, Z. Xu, and R. Akerkar, "Ondevice learning systems for edge intelligence: A software and hardware synergy perspective," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11916–11934, 2021.
- [24] J. Guo, "Learning robust visual-semantic mapping for zero-shot learning," arXiv preprint arXiv:2104.05668, 2021.
- [25] Y. Wang, J. Guo, S. Guo, W. Zhang, and J. Zhang, "Exploring optimal substructure for out-of-distribution generalization via feature-targeted model pruning," *arXiv preprint arXiv:2212.09458*, 2022.
- [26] S. Ma, J. Guo, S. Guo, and M. Guo, "Position-aware convolutional networks for traffic prediction," arXiv preprint arXiv:1904.06187, 2019.
- [27] J. Guo, "An improved incremental training approach for large scaled dataset based on support vector machine," in *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, pp. 149–157, 2016.
- [28] F. Huo, W. Xu, S. Guo, J. Guo, H. Wang, and Z. Liu, "Procc: Progressive cross-primitive consistency for open-world compositional zero-shot learning," *arXiv preprint arXiv:2211.12417*, 2022.
- [29] Q. Zhou, R. Li, S. Guo, Y. Liu, J. Guo, and Z. Xu, "Cadm: Codecaware diffusion modeling for neural-enhanced video streaming," arXiv preprint arXiv:2211.08428, 2022.
- [30] J. Guo and S. Guo, "Ams-sfe: Towards an alignment of manifold structures via semantic feature expansion for zero-shot learning," in 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 73–78, IEEE, 2019.
- [31] Y. Wang, S. Guo, J. Guo, W. Zhang, Y. Xu, J. Zhang, and Y. Liu, "Efficient stein variational inference for reliable distribution-lossless network pruning," arXiv preprint arXiv:2212.03537, 2022.
- [32] J. Guo and S. Guo, "Ee-ae: An exclusivity enhanced unsupervised feature learning approach," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3517–3521, IEEE, 2019.
- [33] Q. Zhou, S. Guo, Z. Qu, J. Guo, Z. Xu, J. Zhang, T. Guo, B. Luo, and J. Zhou, "Octo: Int8 training with loss-aware compensation and backward quantization for tiny on-device learning.," in USENIX Annual Technical Conference, pp. 177–191, 2021.
- [34] J. Guo, S. Guo, J. Zhang, and Z. Liu, "Fed-fsnet: Mitigating noniid federated learning via fuzzy synthesizing network," arXiv preprint arXiv:2208.12044, 2022.
- [35] B. Li, T. Xi, G. Zhang, H. Feng, J. Han, J. Liu, E. Ding, and W. Liu, "Dynamic class queue for large scale face recognition in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*,

*CVPR 2021, virtual, June 19-25, 2021*, pp. 3763–3772, Computer Vision Foundation / IEEE, 2021.

- [36] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, L. Kong, and M. Li, "Lip readingbased user authentication through acoustic sensing on smartphones," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 447–460, 2019.
- [37] K. Wong, Q. Zhang, M. Liang, B. Yang, R. Liao, A. Sadat, and R. Urtasun, "Testing the safety of self-driving vehicles by simulating perception and prediction," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVI*, vol. 12371 of *Lecture Notes in Computer Science*, pp. 312– 329, Springer, 2020.
- [38] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, MobiCom 2018, New Delhi, India, October 29 - November 02, 2018*, pp. 289–304, ACM, 2018.
- [39] H. Zhang, C. Song, A. Wang, C. Xu, D. Li, and W. Xu, "Pdvocal: Towards privacy-preserving parkinson's disease detection using nonspeech body sounds," in *The 25th Annual International Conference on Mobile Computing and Networking, MobiCom 2019, Los Cabos, Mexico, October 21-25, 2019*, pp. 16:1–16:16, ACM, 2019.
- [40] Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, vol. 54 of Proceedings of Machine Learning Research, PMLR, 2017.
- [41] X. Tang, S. Guo, and J. Guo, "Personalized federated learning with contextualized generalization," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2241– 2247, 2022.
- [42] M. Tang, X. Ning, Y. Wang, J. Sun, Y. Wang, H. H. Li, and Y. Chen, "Fedcor: Correlation-based active client selection strategy for heterogeneous federated learning," in *IEEE/CVF Conference on Computer Vision* and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022, pp. 10092–10101, IEEE, 2022.
- [43] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, 2021.
- [44] V. C. Gogineni, S. Werner, Y.-F. Huang, and A. Kuh, "Communicationefficient online federated learning framework for nonlinear regression," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics*, *Speech and Signal Processing (ICASSP)*, pp. 5228–5232, 2022.
- [45] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, London, United Kingdom, May 2-5, 2022*, pp. 1449–1458, IEEE, 2022.
- [46] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.
- [47] Y. Wang, J. Wang, W. Zhang, Y. Zhan, S. Guo, Q. Zheng, and X. Wang, "A survey on deploying mobile deep learning applications: A systemic and technical perspective," *Digital Communications and Networks*, 2021.
- [48] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems -Volume 1*, NIPS'15, (Cambridge, MA, USA), 2015.
- [49] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," J. Emerg. Technol. Comput. Syst., 2017.
- [50] F. M. Rueda, R. Grzeszick, and G. A. Fink, "Neuron pruning for compressing deep networks using maxout architectures," in *Pattern Recognition - 39th German Conference, GCPR 2017, Basel, Switzerland, September 12-15, 2017, Proceedings*, Lecture Notes in Computer Science, 2017.
- [51] Z. Li, B. Ni, W. Zhang, X. Yang, and G. Wen, "Performance guaranteed network acceleration via high-order residual quantization," in 2017 IEEE International Conference on Computer Vision (ICCV), Oct 2017.
- [52] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," *Computer Science*, pp. 2285–2294, 2015.
- [53] V. Smith, C. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multitask learning," in Advances in Neural Information Processing Systems

30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 4424–4434, 2017.

- [54] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, 2016.
- [55] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in 2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019, pp. 2512– 2520, 2019.
- [56] R. Zeng, S. Zhang, J. Wang, and X. Chu, "Fmore: An incentive scheme of multi-dimensional auction for federated learning in MEC," *CoRR*, 2020.
- [57] Q. Wang, S. Shi, C. Wang, and X. Chu, "Communication contention aware scheduling of multiple deep learning training jobs," *CoRR*, 2020.
- [58] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
- [59] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, pp. 12:1–12:19, 2019.
- [60] J. Song, M. H. Oh, and H. Kim, "Personalized federated learning with server-side information," *IEEE Access*, vol. 10, pp. 120245–120255, 2022.
- [61] M. Mortaheb, C. Vahapoglu, and S. Ulukus, "Personalized federated multi-task learning over wireless fading channels," *Algorithms*, vol. 15, no. 11, p. 421, 2022.
- [62] V. Mugunthan, E. Lin, V. Gokul, C. Lau, L. Kagal, and S. D. Pieper, "FedItn: Federated learning for sparse and personalized lottery ticket networks," in *Computer Vision - ECCV 2022 - 17th European Conference*, *Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XII*, vol. 13672 of *Lecture Notes in Computer Science*, pp. 69–85, Springer, 2022.
- [63] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," arXiv preprint arXiv:1806.00582, 2018.
- [64] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *CoRR*, vol. abs/1910.03581, 2019.
- [65] H. Cai, C. Gan, T. Wang, Z. Zhang, and S. Han, "Once-for-all: Train one network and specialize it for efficient deployment," in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020.
- [66] X. Wang, F. Yu, Z. Dou, T. Darrell, and J. E. Gonzalez, "Skipnet: Learning dynamic routing in convolutional networks," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, pp. 420–436, 2018.