# FedCKD: KD-based Classifier Refinement for Non-IID Data in Federated Learning

Chao Xu School of Computer Science, Nanjing Audit University, Nanjing, China xuchao@nau.edu.cn

Wenzhang Su School of Computer Science, Nanjing Audit University, Nanjing, China mp2309041@stu.nau.edu.cn

Fanfan Shen School of Computer Science, Nanjing Audit University, Nanjing, China ffshen@whu.edu.cn

# Jun Zhang

Yanxiang He School of Computer Science, Wuhan University,Wuhan yxhe@whu.edu.cn

# Yong Chen

School of Software, East China University of Science and Technology, Nanchang 664633281@qq.com

School of Computer Science, Nanjing Audit University, Nanjing, China chenyong@nau.edu.cn

Abstract —In federated learning, client data typically exhibits non-independent and identically distributed (non-IID) character- istics, which introduce classifier bias across clients and degrade global model performance. To tackle this challenge, we propose FedCKD (Federated Classifier Refinement via Knowledge Distil- lation), a framework that mitigates data heterogeneity by refining the global classifier on the server through knowledge distillation. Specifically, each client fits its local feature distribution with a multi-component Gaussian Mixture Model (GMM) and uploads the parameters to the server. The server then generates virtual feature samples from the aggregated statistics, performs weighted aggregation of soft labels to construct teacher vectors, and finally applies a decoupled knowledge distillation mechanism to refine the global classifier. The experimental results show that under the most severe data heterogeneity setting on the CIFAR-10 dataset, FedCKD outperforms state-of-the-art algorithms by ${\bf6.93\%}$ and in accuracy. Moreover, it also outperforms other baseline methods under non-IID settings across different datasets.

CCVR (Classifier Calibration with Virtual Representations) [6] identifies the classifier layer bias as a primary reason for performance degradation in federated learning with non- IID data. It highlights the low feature similarity among local classifiers across clients, along with weight norms that dis- proportionately favor majority classes. Consequently, classi- fier optimization becomes essential for improving federated learning performance under data heterogeneity. Knowledge distillation [7], as a model compression and transfer learn- ing technique, has recently been widely adopted to miti- gate data heterogeneity in FL. Based on the findings of CCVR, we propose FedCKD (Federated Classifier Refinement via Knowledge Distillation). Unlike CCVR, which performs post hoc calibration using single-Gaussian virtual features for each class, FedCKD continuously refines the classifier during training. It captures aggregated statistical knowledge via a multi-component Gaussian Mixture Model (GMM), and applies class-aware knowledge distillation guided by client label distributions.

Index Terms —Federated Learning, Non-IID Data, Knowledge Distillation, Classifier Refinement

# I. I NTRODUCTION

Federated learning (FL) is an emerging distributed machine learning paradigm that enables clients to collaboratively train models while maintaining data privacy [1] [2]. However, data across clients in FL is typically non-independent and identically distributed (non-IID), exhibiting imbalanced sam- ple sizes, heterogeneous feature spaces and skewed label distributions. To tackle this issue, numerous studies have been conducted. Scaffold (Stochastic Controlled Averaging for Federated Learning) [3] mitigates client drift caused by data heterogeneity, using control variates on the server to correct the direction of client updates. FedNova (Federated Normalized Averaging) [4] enables flexible hyperparameter selection on the server and promotes consistent global model convergence by reweighting client contributions during ag- gregation. Instead of traditional model averaging, FedConcat (Federated Learning with Model Concatenation) [5] adopts a model concatenation framework that clusters clients, trains feature extractors within each cluster, and jointly trains a global classifier to mitigate the impact of data heterogeneity.

Specifically, our main contributions are as follows:

We propose a virtual feature generation method in which • each client models its local features using a multi- component GMM, and the server aggregates them in two stages to construct a global per-class GMM. Based on this, class-balanced synthetic features are generated to calibrate the global classifier. We design a class-aware teacher vector by weighting • client soft logits with label distributions, and refine the global classifier using a decoupled distillation loss that separately supervises target and non-target classes to mitigate inter-client bias. We conduct extensive experiments on the CIFAR-10, • CIFAR-100, CINIC-10 and MNIST datasets under vary- ing degrees of data heterogeneity. The results show that FedCKD consistently outperforms baseline methods and exhibits strong generalization ability, especially under severe non-IID settings on the CIFAR-10.

Figure 1: The framework of FedCKD. In FedCKD, clients not only update their local model parameters to , but also collect extra statistical knowledge to upload to the server. The server leverages this knowledge to perform decoupled knowledge distillation training on the global classifier. (The elements in the framework are obtained from iconfont.)

# II. PROPOSED METHOD

# A. Federated Learning Setup

We consider a federated learning system consisting of clients and a central server. Each client holds a local dataset , where is the input data, is the corresponding label and is the number f samples in the local dataset. In each communication round t , the server ran- domly selects clients to participate in collaborative training. The selected clients receive the global model parameters broadcast by the server and perform local training epochs. Each client updates the local model parameters to based on its local data and uploads it to the server for aggregation. The objective of can be expressed as in [1]:

where is the local loss function of client

Since the classifier is directly influenced by the local label distribution, classifiers on different clients can exhibit significant bias and inconsistency in non-IID scenarios [6]. In addition to standard training, clients are required to provide extra knowledge to assist in refining the classifier. The Fig. 1 illustrates the design framework of the FedCKD algorithm. A detailed implementation is presented in the following section.

B. Client Training

For each participating client , in addition to performing standard local training to update its local model to , it is also required to collect extra local knowledge after each local training phase. This knowledge includes feature distribution statistics and the statistical information of classifier outputs.

We assume that the entire dataset consists of classes. For each sample in class , the local model extracts a feature vector , where and denotes the number of samples n class on client . If 1 ), meaning that class c has sufficient samples, we model its feature distribution using a multi-component GMM as follows:

where denotes a Gaussian distribution, is the num- ber of Gaussian components used for class , represents e weight of the -th com ). The vector of the is he mean vector and -th component. is the variance

When , the number of samples in class is insufficient to reliably fit a multi-component GMM. In this case, a single Gaussian distribution is used to model the class, and the sample mean and variance for each feature dimension are computed as follows:

To support the construction of teacher vectors on the server, local model is also required to upload soft label knowledge from the classifier outputs. Specifically, we first compute the average logits for class as follows:

where denotes the classifier output of client for class . To further capture the uncertainty in the outputs, the variance of the logits for each class is also uploaded:

# C. Server Aggregation and Refinement

Upon receiving the additional statistical knowledge up- loaded by clients, the server aggregates it to construct teacher vectors and a virtual feature dataset. To generate virtual fea- tures for each class , we perform a two-stage aggregation to preserve the mixture structure. Firstly, if the GMMs uploaded by clients for the class have the same number of components, we align them by sorting local GMM components, and use the client sample sizes as weights to obtain the global statistics:

where and are the mean, diagonal variance and mixture weight of the -th component in the aggregated distribution for class .

If the same class appears with different component counts among clients, we first perform the aggregation steps (7)- (9) within each component-count group separately, obtaining distinct aggregated sub-GMMs. Next, we concatenate these sub-GMMs to form a unified multi-component GMM. To properly handle the differences in component counts, we scale the mixture weights of each sub-GMM according to the total number of samples from the corresponding client groups. After concatenation, the mixture weights are re-normalized to en ure that , where is the total number of components kept for class . The global distribution for class c is then represented by a unified multi-component GMM defined as:

If all clients upload single-component Gaussian statistics for a given class, the server aggregates them by comput- ing the standard mean and variance weighted by the client sample sizes . To generate virtual features for class , we first sample a component index

(categorical distribution), and then sample a feature vector . Assuming independence between feature dimensions, the density of the sampled virtual feature is expressed as:

The server randomly generates $({N^{\*}}\mathrm{~\ensuremath~{~>~}~}0)$ ) virtual features for each class based on the predefined number of virtual samples. To construct a class-aware teacher, we first weighted fusion, and the aggregated mean and variance are utilize the class-aware weights P to perform obtained as follows:

Accordingly, the aggregated per-class logit is modeled as:

The server performs multi-view sampling for each class by drawing ) samples ∼ from the corresponding normal distribution N . Each sampled value is then processed with temperature scaling followed by a softmax operation. The final teacher vector is defined as:

where is the temperature parameter.

During the knowledge distillation process, the teacher vector is decoupled into target and non-target components. For each virtual sample , the server uses the current global model’s classifier as the student model to perform a forward pass and obtain the output . When supervising the target class component, the distribution of the student model is given by:

For the target class label , the corresponding distribution is denoted as . As it primarily reflects hard label super- vision, the target loss is computed via cross-entropy:

For the non-target class, in order to transfer the dark knowl- edge that reflects class-level distinctions in a fine-grained manner, temperature scaling is applied to the output of the student model. The resulting distribution of the student model is given by:

|  |
| --- |
| TABLE I: Comparison of Top-1 accuracy of various algorithms under non-IID data |

TABLE I: Comparison of Top-1 accuracy of various algorithms under non-IID data

Accordingly, Kullback-Leibler (KL) divergence is used to align the outputs of the teacher and student models:

The overall loss for distillation on a virtual sample is then computed as the weighted sum of the target and non-target components:

where is used to balance the two components of the loss.

The server uses the virtual feature dataset to supervise the global classifier, optimizing it via multiple epochs of gradient descent guided by the total distillation loss. After epochs of training, the distilled and refined global feature classifier is obtained and broadcast to the clients to initiate the next round of federated training.

# III. EXPERIMENTS

# A. Experimental Setup

Dataset. Our experiments are conducted on four datasets: CIFAR-10, CIFAR-100, CINIC-10 and MNIST, with CIFAR- 10 serving as the primary dataset. We adopt the Latent Dirich- let Allocation (LDA) strategy to simulate data heterogeneity among clients, where the parameter controls the degree of heterogeneity. As decreases, the degree of data heterogeneity increases. On the CIFAR-10 dataset, we set to 0.05, 0.1, 0.3 and 0.5. For the other datasets, is fixed at 0.1. Our experiments are implemented in Python 3.8 using the PyTorch framework (version 1.10.0 with CUDA 11.3), and executed on a machine equipped with an NVIDIA RTX 3090 GPU and Intel(R) Xeon(R) Gold 6330 CPU .

Baseline. FedCKD is fairly compared against five baselines: FedAvg [1], FGA [2], Scaffold [3], FedConcat [5] and CCVR [6]. For all algorithms, we use a lightweight CNN model consisting of two convolutional and pooling layers. After feature extraction, the classifier includes a fully connected layer that maps the 512-dimensional feature vector to the number of classes, with ReLU activation and dropout applied.

Implementation. For the federated learning task, we set the total number of clients to , and randomly select clients to p te in training during each round. Each client performs local training epochs based on its own data, and the total number of communication rounds is set to . Specifically, for CCVR and FedCKD, the number of additional server-side refinement steps is set to , and the number of virtual samples generated per class is . For all methods, we use Stochastic Gradient Descent (SGD) to optimize the classifier, with a learning rate of 0.01, momentum of 0.9 and weight decay of 1e-5. Hyperparameters for all methods are carefully selected through grid search to achieve optimal results across all experiments.

# B. Experimental Results and Analysis

Tab. I presents the performance of various algorithms un- der different settings across four datasets, using Top-1 test accuracy as the evaluation metric. On the CIFAR-10 dataset with the most severe non-IID setting ( ), FedCKD outperforms CCVR and FedConcat with accuracy improve- ments of and , respectively. Notably, the Scaffold fails to converge when . As increases, all methods show a consistent rise in accuracy, with FedCKD maintaining a leading position throughout. Notably, when and data heterogeneity is no longer the primary bottleneck, the FedConcat algorithm achieves the highest accuracy. However, FedCKD still maintains strong competitiveness and is on a par with the best-performing baseline.

The experimental results under varying degrees of data heterogeneity show that FedCKD performs best with moderate to high heterogeneity . Its effectiveness stems from the ability to generate complementary virtual features for minority classes and construct multi-view teacher vectors. A decoupled distillation loss further fine-tunes the global classifier by aligning hard-label boundaries with nuanced dark knowledge. FedCKD mitigates weight inflation in majority classes while enhancing minority class discrimination, without causing negative transfer as heterogeneity decreases.

Meanwhile, FedCKD achieves the best results on the CIFAR-100 and CINIC-10 datasets ), outperforming CCVR by and , respectively. However, on the simpler MNIST handwritten digit dataset ), FedCKD ranks second but still remains competitive with the best- performing baseline.

# C. Discussion

To further support the experimental analysis, we investigate three key hyperparameters: the number of local training epochs on the client side , the number of refinement steps on the server side and the number of virtual samples generated per class . All experiments in this section are conducted on the CIFAR-10 dataset with .

We evaluate values of 1, 5, 10 and 20, with results shown in Fig. 2. As increases, the accuracy of all methods consistently improves. This is attributed to more sufficient local gradient updates and reduced communication frequency. When , the performance gap between algorithms narrows significantly, yet FedCKD continues to outperform all baselines. The results demonstrate that FedCKD is robust to the number of local training epochs. However, a higher also brings significant computational overhead per communication round for all methods.

Figure 2: Accuracy comparison under different local training epochs

Since both FedCKD and CCVR refine the classifier on the server side, we compare them by varying the number of server-side training epochs . The results are shown in Tab. II. It can be observed that the accuracy of both methods improves with an increasing number of refinement steps. For CCVR, the gain becomes marginal at , and performance even drops at . This is because the training dataset is generated from sampled features, which introduces some noise. Increasing the number of training epochs excessively amplifies the impact of this noise in the dataset. With the aid of knowledge distillation, FedCKD does not experience a significant accuracy decline at , effectively mitigating the negative impact of noise.

|  |
| --- |
| TABLE II: Accuracy comparison under different server-side training epochs |

TABLE II: Accuracy comparison under different server-side training epochs

|  |
| --- |
| TABLE III: Comparison of accuracy under varying virtual sample sizes |

TABLE III: Comparison of accuracy under varying virtual sample sizes

Another key factor we examine is the number of virtual features generated, as shown in Tab. III. As the number of generated virtual features increases, the accuracy of CCVR and FedCKD improves. When the number of virtual features generated by FedCKD is , its accuracy approaches that of CCVR with . This demonstrates that knowl- edge distillation for refinement can achieve better accuracy with fewer virtual features.

# IV. CONCLUSION

In this paper, we propose FedCKD, a federated learning framework designed to address data heterogeneity by refining the global classifier on the server side via knowledge distilla- tion. FedCKD models local feature distributions using a multi- component Gaussian Mixture Model and constructs class- aware teacher vectors on the server. By decoupling knowledge distillation, it effectively calibrates classifier bias. Experi- mental results show that FedCKD consistently outperforms prior methods in both accuracy and robustness. Moreover, by generating virtual features instead of transmitting raw data, it effectively preserves data privacy. In future work, we aim to scale FedCKD to broader deployment scenarios and further reduce communication overhead.

# A CKNOWLEDGMENT

This work has been supported by the Basic Science (Natural Science) Research Project of Colleges and Universities in Jiangsu Province (22KJA520004, 24KJA520005), the National Natural Science Foundation of China (61902189, 62472227, 62162002, 61972293), Postgraduate Research & Practice In- novation Program of Jiangsu Province (KYCX25 2479).

# R EFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in Artificial intelligence and statistics . PMLR, 2017, pp. 1273– 1282.

[2] S. W. Remedios, J. A. Butman, B. A. Landman, and D. L. Pham, “Federated gradient averaging for multi-site training with momentum- based optimizers,” in Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning . Springer, 2020, pp. 170– 180.

[3] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” in International conference on machine learning . PMLR, 2020, pp. 5132–5143.

[4] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the ob- jective inconsistency problem in heterogeneous federated optimization,” Advances in neural information processing systems , vol. 33, pp. 7611– 7623, 2020.

[5] Y. Diao, Q. Li, and B. He, “Exploiting label skews in federated learning with model concatenation,” in Proceedings of the AAAI Conference on Artificial Intelligence , vol. 38, no. 10, 2024, pp. 11 784–11 792.

[6] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, “No fear of heterogeneity: Classifier calibration for federated learning with non-iid data,” Advances in Neural Information Processing Systems , vol. 34, pp. 5972–5984, 2021.

[7] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, “Decoupled knowledge distillation,” in Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition , 2022, pp. 11 953–11 962.