Model Drift via Neural Collapse and Negative Label Knowledge Distillation

Sen Yu[1], Ruizhi Pu[2], Shaojie Zhan[3], Xiuting Weng[1], Yuanhang Yao[4], Lixing Yu[*1]
[1] School of Information Science and Engineering, Yunnan University, Kunming, China
[2] School of Statistics and Data Science, Southeast University, Nanjing, China
[3] Department of Computer Science, Texas Tech University, Lubbock, USA
[4] School of Public Health, Southeast University, Nanjing, China
Contact: yulixing@ynu.edu.cn

***Abstract*—Federated learning (FL) is severely impacted by model drift under data heterogeneity, manifesting as simultaneous feature representation drift and classifier drift across local clients—a dual challenge that collectively impairs global model aggregation performance. Current methodologies predominantly address a singular aspect of this challenge. In pursuit of a comprehensive resolution, we introduce a novel framework grounded in neural collapse theory. Our framework aims at achieving thorough drift mitigation by decoupling features from classifiers. We replace cross-entropy loss with hyperspherical uniformity gap loss in FL frameworks to mitigate classifier bias through data-agnostic means. Additionally, we reconstruct latent negative label knowledge distributions using knowledge distillation to ensure geometric consistency in feature representations across diverse clients. Extensive experiments validate the effectiveness of the proposed method in heterogeneous data.**

***Index Terms*—data heterogeneity, neural collapse, knowledge distillation**

I. Introduction

Federated learning (FL) [1] enables collaborative model training while preserving data privacy. Canonical FL frameworks, exemplified by FedAvg [2], excel in performance with independently and identically distributed (IID) data through the averaging local model parameters [3]. However, in real-world settings with non-independent and identically distributed (Non-IID) data [4] characterized as data heterogeneity, a phenomenon known as model drift arises, involving concurrent feature representation misalignment and classifier discrepancy within local models. As illustrated empirically in Fig. 1, data heterogeneity triggers both types of drift, leading to a decline in global model performance during aggregation.

Contemporary methodologies [5], [6] predominantly concentrate on tackling discrete facets of this dual challenge, resulting in partial and insufficient resolutions. Drawing insights from the neural collapse (NC) theory [7], positing that adept models gravitate towards equiangular tight frame (ETF) configurations to enhance feature representation and classifier alignment, our proposition entails a strategy centered on segregating feature representation from classifiers to rectify the shortcomings in extant approaches.

Nevertheless, data heterogeneity in FL fundamentally impedes NC convergence, manifested by classifiers biased toward local label distributions and feature representations that neglect negative label knowledge. This dual impairment is exemplified when a client possessing only 'cat' data fails to maintain geometric consistency for absent classes such as 'bird'. Such dual impairment is challenging to maintain geometric consistency in clients when confronted with data labels they do not possess.

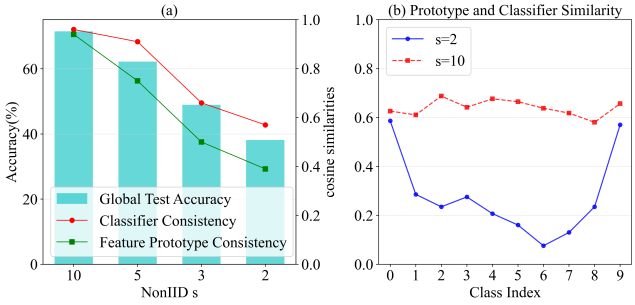

Fig. 1. The experiment utilizes basic FedAvg on the CIFAR-10 dataset, where s serves as the partitioning parameter regulating data segmentation. A lower s implies reduced labels per client and heightened data heterogeneity. (a) With escalating data diversity, the global model's accuracy diminishes, accompanied by a decrease in cosine similarity between the global model's feature prototypes and classifier weights compared to those of the local model. (b) The cosine similarity between the global model's feature prototypes and classifier weights decreases under Non-IID data conditions at $s = 2$.

To address these limitations, we propose a dual strategy framework that combines the hyperspherical uniformity gap (HUG) loss [8] with negative label knowledge distillation (KD). HUG loss enhances the resilience of ETF structured classifiers against data skewness, effectively mitigating classifier drift caused by Non-IID data distributions. Meanwhile, KD reconstructs latent negative label distributions to ensure consistency in cross-client features, effectively addressing feature representation drift. This unified methodology marks the inaugural utilization of HUG in decentralized settings, yielding cutting-edge performance in accuracy and convergence speed. Code will be made publicly available upon acceptance.

II. Related Work

A. Data Heterogeneity in Federated Learning

To address the critical issue of data heterogeneity in FL, existing approaches typically involve carefully optimizing local

* Corresponding author.

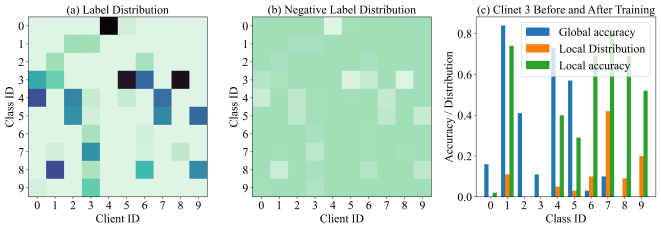

Fig. 2. (a) The local label distribution of clients. (b) The negative label distribution. (c) The accuracy rate of client 3 before and after local training.

models [9] and global models [10]. Building upon FedAvg, FedProx [11] introduces a proximal term to regularize local model updates, mitigating client drift. FedACG [12] integrates the momentum term into the model parameters, making the local update closer to the global gradient direction and reducing drift. FedMut [13] optimizes the global model through a gradient-guided mutation strategy, enabling local models to converge toward flat regions of the loss function. FedFN [14] eliminates the feature norm difference between the local model and the global model by introducing feature normalization in the local training on the client side.

B. Neural Collapse

The NC was initially observed during the terminal phase of model training. During the neural collapse process, the weights of the last layer collapse to the class-wise mean of features, the features of each class collapse to this class-wise mean, and these class-wise mean of features further collapse into a simple ETF. Subsequently, several studies [15], [16] investigated the causes of neural collapse and elucidated its underlying principles. FedETF [17] synthesized and fixed ETF classifiers to address the bias issue in FL classifiers.

C. Knowledge Distillation in Federated Learning

KD [18] transfers knowledge from a cumbersome teacher model to a lightweight student model, thereby providing an effective mechanism to mitigate data heterogeneity in FL through parameterized knowledge harmonization [19]. FedHKD [20] shares the client's hyper knowledge, which contains the class prototype and the soft prediction, to align the client's training direction. FedNTD [21] prevents global knowledge forgetting by distilling only the knowledge of not-true classes during local training on the client side.

III. Neural Collapse Principles in FL

A. Problem Statement

We consider a typical FL system involving K clients and one server. C denotes the global data classes. The client k's dataset is $\mathcal{D}^k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ ($\left|\mathcal{D}^k\right| = N_k$). The trained model, parameterized by feature extractor θ and classifier ϕ, has full parameters $w = \{\theta, \phi\}$. In each communication round t, the server samples K^t clients and distributes the current global parameters w_g^{t-1}. The sampled clients update their parameters as $w_k^t = w_g^{t-1}$ and conduct training using their respective datasets. Following local training, K^t clients transmit their parameters to the server for aggregation, typically employing the FedAvg.

$$w_g^t = \sum_{k \in K^t} \frac{N_k}{\sum_{k' \in K^t} N_{k'}} w_k^t. \tag{1}$$

B. HUG Loss for Decoupled Optimization

The NC theory establishes dual principles for deep learning: maximizing inter-class separability while minimizing intra-class variance. Nevertheless, the traditional cross-entropy (CE) loss does not explicitly optimize these two objectives, leading to suboptimal training outcomes in FL. To this end, we decouple the model training for integrating NC by introducing the HUG loss [8] as

$$L_{HUG} = \alpha \cdot \sum_{c \neq c'} \left\| \widehat{\phi}_c - \widehat{\phi}_{c'} \right\|^{-2} + \beta \cdot \sum_c \sum_{i \in A_c} \|\widehat{z}_i - \widehat{\phi}_c\|, \tag{2}$$

where α and β are hyper-parameters, $\widehat{\phi}_c = \frac{\phi_c}{|\phi_c|}$ normalizes the classifier weights of class c, $\widehat{z}_i$ normalizes the features extracted by the feature extractor from input data x, and A_c represents the data for class c. The first term promotes uniform classifier distribution on a hypersphere for enhanced inter-class separation, while the subsequent term tightens feature clusters around class prototypes to ensure intra-class compactness.

IV. Proposed Method

Our framework integrates NC theory with negative label KD to handle data heterogeneity in FL. By decoupling the optimization of positive and negative label relationships, our core innovation allows local models to specialize in local features while retaining global feature knowledge concurrently.

A. Local Label Distribution Analysis

In FL with Non-IID data, each client k possesses a local label distribution $p_k = (p_k^1, \ldots, p_k^C)$ where $p_k^c = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{I}(c = y_i^k)$. This distribution is typically skewed, with certain classes entirely absent from local data. To counteract this distribution bias, we introduce the negative label $\tilde{y}_i^k = \left\{c \in C \mid c \neq y_i^k\right\}$ for sample (x_i^k, y_i^k). The negative label distribution is as follows $\tilde{p}_k = (\tilde{p}_k^1, \ldots, \tilde{p}_k^C)$ where $\tilde{p}_k^c = \frac{1}{(C-1)N_k} \sum_{i=1}^{N_k} \mathbb{I}(c \in \tilde{y}_i^k)$. It can also be called out-of-distribution knowledge.

The local distribution and its negative label distribution corresponding to the client are shown in Fig. 2(a) and Fig. 2(b). It can be seen that labels with a higher proportion in the label distribution will have a lower proportion in the negative label distribution. In Fig. 2(c), after Client 3 is locally trained using HUG, the prediction accuracy of the local model for minority classes and missing classes will decrease, while the accuracy for majority classes will increase. Therefore, local training should retain the knowledge of negative labels.

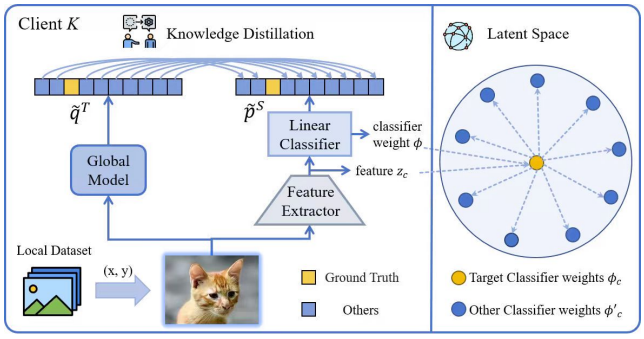

Fig. 3. Our framework employs a dual-loss mechanism for client training: The HUG loss aligns local features with classifier weights, enforcing the ETF structure on the weight matrix to prevent classifier drift. The negative label KD loss preserves negative label knowledge by aligning the local negative label distribution with that of the global model, thereby mitigating feature drift effectively.

B. Negative Label Knowledge Distillation

We utilize HUG loss for enforcing intra-class compactness and inter-class separation in local positive labels. To prevent catastrophic forgetting of global knowledge and mitigate feature drift, we apply KD to negative labels. The framework is depicted in Fig. 3.

Initially, for each sample (x, y) within the local client's dataset, we derive its label y and calculate the negative labels $\tilde{y} = \{c \in C \mid c \neq y\}$. The local model, under HUG loss, is susceptible to forgetting such negative label information. To address this issue, we designate the local model as the student and the global model as the teacher. By leveraging solely the local client data, we facilitate the local model to assimilate the knowledge of negative labels through KD. The negative label knowledge of the server model is formally defined as

$$\tilde{q}^T(c \in \tilde{y}|x) = \frac{exp(Y_c/\tau)}{\sum_{c' \in \tilde{y}} exp(Y_{c'}/\tau)}, \tag{3}$$

where Y_c represents the logit value of class c within the server model's output, while τ signifies the temperature coefficient governing the knowledge of the smoothed label.

To enhance the client model's classifier performance, the HUG loss is instrumental. Consequently, we separate model outputs and exclusively adjust the gradients of feature representations. Additionally, as model depth increases, discrepancies among models trained on diverse data distributions become more pronounced. Hence, feature normalization is implemented to mitigate these biases in local models. The negative label knowledge of the local model is formally characterized as

$$\tilde{p}^S(c \in \tilde{y}|x) = \frac{exp(\hat{z}_x \cdot \phi_c/\tau)}{\sum_{c' \in \tilde{y}} exp(\hat{z}_x \cdot \phi_{c'}/\tau)}, \tag{4}$$

where $\hat{z}_x = \frac{z_x}{|z_x|}$ signifies the normalized version of the feature representation z_x corresponding to data x, and ϕc denotes the classifier weights associated with class c in the local model.

Algorithm 1

Input K clients' dataset; global model parameters θ_g and ϕ_g; T: global rounds; E: local epochs; α, β, μ and τ: hyper-parameter; η: local learning rate
Initialize global model parameters θ_g^0 and ϕ_g^0
for each communication round $t = 1, \cdots, T$ **do**
  select client set K^t
  for each client $k \in K^t$ **do**
    set model parameters $\theta_k^{t,0} = \theta_g^{t-1}$, $\phi_k^{t,0} = \phi_g^{t-1}$
    for each local epoch $e = 1, \cdots, E$ **do**
      $\theta_k^{t,e} = \theta_k^{t,e-1} - \eta \nabla_\theta L_{HNKD}(\theta_k^{t,e-1}, \theta_g^{t-1}, D_k)$
      // Using Eq. (6)
      $\phi_k^{t,e} = \phi_k^{t,e-1} - \eta \nabla_\phi L_{HUG}(\phi_k^{t,e-1})$
      // Using Eq. (2)
    end for
  end for
  Upload θ_k^t and ϕ_k^t to server // Using Eq. (1)
end for

The modified model distillation is formulate as follows

$$L_{NKD}(\tilde{p}^S, \tilde{q}^T) = -\sum_{c}^{\tilde{y}} \tilde{q}^T(c|x) \log \left[\frac{\tilde{p}^S(c|x)}{\tilde{q}^T(c|x)} \right]. \tag{5}$$

To this end, the comprehensive training objective is delineated as follows

$$L_{HNKD} = L_{HUG} + \mu \cdot \tau^2 \cdot L_{NKD}(\tilde{p}^S, \tilde{q}^T), \tag{6}$$

where μ is a hyper-parameter related to the distillation weight.

By exclusively adapting client training methodologies within Federated Learning, our approach sidesteps the introduction of supplementary communication overhead or privacy vulnerabilities, concurrently enhancing model efficacy. The exhaustive training procedure is explicitly elucidated in Algorithm 1.

C. Convergence Analysis

If the learning rate is chosen as $\eta = \frac{1}{L\sqrt{T}}$ with T being the total number of communication rounds, the global model satisfies:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\|\nabla F(w_g^t)\|^2\right] \leq \frac{2L\big(F(w^0) - F^*\big)}{\sqrt{T}} + \frac{4L^2\eta^2E^2(\sigma^2 + \zeta^2)}{T} + \frac{\sigma^2}{K\sqrt{T}}, \tag{7}$$

where $F^* = \inf_w F(w)$ is the global minimum, w^0 is the initial parameter, L, σ and ζ are constants greater than zero. As $T \to \infty$, the right-hand side approaches zero, ensuring convergence.

V. Experiments

A. Experimental Setups

1) Baselines: Our approach aims to enhance the performance of FL in scenarios with heterogeneous data. Therefore, we primarily compare the traditional FedAvg algorithm

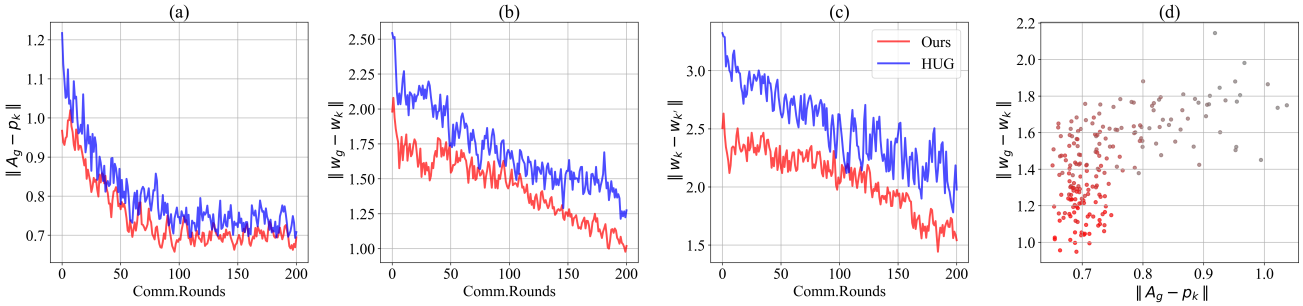

Fig. 4. Comparison of weights and distributions difference between HUG and our method. (a) The distance between distributions. (b) The differences between the global model and local models. (c) The differences among local models. (d) The relationship between weights and distributions, opacity encodes later communication rounds.

with several algorithms tailored to address data heterogeneity at the client level, including FedProx [11], FedNTD [21], FedFN [14], FedETF [17], FedMut [13] and FedACG [12].

2) Datasets: In the experiments of this paper, five widely used datasets in FL were utilized: MNIST, FashionMNIST, CIFAR-10, CIFAR-100 and SVHN. To simulate a Non-IID data environment, we employ two data partitioning methods. Sharding: In the dataset, each class is divided into multiple slices, each containing an equal amount of data. The heterogeneity of client data increases as the number of slices decreases. We set the parameter s for the datasets as follows: MNIST ($s = 2$), FashionMNIST ($s = 2$), CIFAR-10 ($s = 2, 3, 5, 10$), CIFAR-100 ($s = 20$) and SVHN ($s = 2$). Dirichlet Allocation: This method generates Non-IID data through Dirichlet sampling. The smaller the parameter dir, the greater the heterogeneity of the data distribution. We set the parameter dir for the datasets as follows: MNIST ($dir = 0.1$), FashionMNIST ($dir = 0.1$), CIFAR-10 ($dir = 0.05, 0.1, 0.3, 0.5, 1.0$), CIFAR-100 ($dir = 0.2$) and SVHN ($dir = 0.1$).

3) Implementation Details: Following the configurations of previous studies, we set up our experimental environment as follows: The number of clients was set to 100 with a sampling ratio of 0.1. The number of local training epochs for each client was fixed at 5, and the total number of communication rounds was set to 350. We employ the momentum SGD method with a learning rate of 0.05 and a momentum parameter set to 0.9. The learning rate decreases by a factor of 0.99 per round.

B. Ablation Studies

This section presents a comprehensive ablation study to verify the effectiveness of HUG and negative label KD and quantify their impact on the model. We first conduct controlled experiments isolating the contribution of each core component, followed by geometric analysis demonstrating how our method rectifies the distributional biases inherent in standalone HUG optimization under FL.

First, we compare our method with FedAvg, HUG, HKD (HUG combined with traditional KD), and CNKD (CE combined with negative label KD), with results presented in Table I. We observe that HUG loss surpasses FedAvg in accuracy by separating CE training and refining the classifier. While traditional KD transfers knowledge across all classes to preserve global model information, it constrains the local model from effectively learning positive labels. In contrast, CNKD combines crucial training principles, diminishing distillation effectiveness. Therefore, we employ HUG loss for high performance and negative label knowledge distillation to mitigate feature drift.

TABLE I
THE TEST ACCURACY ON CIFAR-10 WITH DIFFERENT dir.

dir	FedAvg	HUG	HKD	CNKD	Ours
0.05	31.52	33.82	39.23	40.23	**43.29**
0.1	47.02	49.56	54.46	55.23	**57.63**
0.3	60.79	62.49	64.47	64.43	**66.47**

Additionally, to demonstrate the enhancement of our method on HUG, we evaluated the differences in weights and distributions. The results are shown in Fig. 4. Although local data distributions can be arbitrary, the underlying distribution of the most robust global model must unify the data from all clients. We can predict the distribution of the global model using an approximation $A_g = \frac{1}{A}(a_1, \ldots, a_c)$, where A denotes the accuracy of the global model and a_c represents accuracy for class c. When class c has more training data, its accuracy increases, which is reflected in a smaller $\| A_g - p_k \|$. In Fig. 4(a), our method provides a better data distribution for local training, leading to improved model performance. In Fig. 4(b) and Fig. 4(c), we systematically quantify inter-model discrepancies across training rounds. In Fig. 4(d), we observe a positive correlation between weight differences and distributional divergence, suggesting that when the local distribution is closer to the global distributions, the post-training weight shifts of local models also become smaller.

C. Performance on Data Heterogeneity

We demonstrate the effectiveness of our method through two different partitioning strategies, with the results presented in Tab. II. The effectiveness of the proposed method is verified by different data and heterogeneity. As the degree of heterogeneity increases, the performance of the proposed

TABLE II

THE TOP 1 TEST ACCURACY (%) ON MNIST, CIFAR-10, AND CIFAR-100. THE VALUES IN PARENTHESES ARE THE STANDARD DEVIATION OF THE LAST 50 ROUNDS. THE ARROW INDICATE THE PERFORMANCE IMPROVEMENTS OF OTHER ALGORITHMS RELATIVE TO FEDAVG.

Non-IID Partition Strategy: Sharding								
Method	**MNIST**	**FashionMNIST**	**SVHN**	**CIFAR-10** $s=2$	**CIFAR-10** $s=3$	**CIFAR-10** $s=5$	**CIFAR-10** $s=10$	**CIFAR-100**
FedAvg	98.56(0.10)	88.97(0.09)	69.57(0.95)	38.13(0.64)	48.95(0.74)	62.19(0.90)	71.49(0.84)	34.31(0.75)
FedProx	98.62(0.11) ↑	86.74(0.09) ↓	67.98(0.81) ↓	43.56(0.61) ↑	53.23(0.69) ↑	64.32(0.73) ↑	69.92(0.81) ↓	32.36(0.58) ↓
FedNTD	98.61(0.08) ↓	88.65(0.18) ↓	75.61(0.64) ↑	43.86(0.54) ↑	52.37(0.82) ↑	64.32(0.98) ↑	72.51(1.11) ↑	35.79(0.98) ↑
FedFN	98.63(0.08) ↑	89.20(0.11) ↑	68.79(0.63) ↓	40.13(0.83) ↑	53.06(0.40) ↑	62.23(0.85) ↓	71.11(0.84) ↓	34.21(0.75) ↓
FedETF	98.35(0.06) ↓	87.50(0.16) ↓	73.65(0.78) ↑	41.98(1.13) ↑	53.65(0.96) ↑	65.81(0.85) ↑	71.25(0.98) ↓	34.97(0.64) ↑
FedMut	98.58(0.05) ↑	89.12(0.10) ↑	74.79(0.03) ↑	42.86(0.83) ↑	54.37(0.42) ↑	65.23(0.86) ↑	72.51(0.84) ↑	34.89(0.98) ↑
FedACG	98.29(0.16) ↓	88.23(0.12) ↓	72.25(0.92) ↑	44.17(0.75) ↑	52.93(0.77) ↑	64.53(1.16) ↑	72.24(0.87) ↑	35.21(0.75) ↑
Ours	**98.95**(0.09) ↑	**89.85**(0.10) ↑	**78.58**(0.74) ↑	**45.19**(0.42) ↑	**55.72**(0.66) ↑	**67.19**(0.78) ↑	**73.86**(0.82) ↑	**36.43**(0.81) ↑
Non-IID Partition Strategy: LDA								
Method	**MNIST**	**FashionMNIST**	**SVHN**	**CIFAR-10** $dir=0.1$	**CIFAR-10** $dir=0.3$	**CIFAR-10** $dir=0.5$	**CIFAR-10** $dir=1.0$	**CIFAR-100**
FedAvg	98.75(0.29)	88.88(0.09)	80.27(1.35)	47.01(0.90)	60.79(0.86)	66.81(0.68)	69.08(1.08)	34.65(0.79)
FedProx	98.81(0.17) ↑	87.17(0.12) ↓	81.67(1.18) ↑	50.90(1.13) ↑	61.59(0.71) ↑	64.36(0.39) ↓	68.74(0.81) ↓	32.49(0.59) ↓
FedNTD	98.32(0.09) ↓	88.77(0.08) ↓	84.62(0.61) ↑	56.83(0.82) ↑	64.80(0.40) ↑	69.85(0.42) ↑	70.05(1.02) ↑	36.23(0.91) ↑
FedFN	99.06(0.09) ↑	**89.56**(0.12) ↑	79.72(0.86) ↓	54.10(1.01) ↑	65.09(0.46) ↑	69.92(1.18) ↑	70.26(1.03) ↑	35.69(0.79) ↑
FedETF	99.01(0.06) ↑	89.04(0.12) ↑	82.51(1.13) ↑	55.57(0.76) ↑	64.16(0.84) ↑	68.87(0.62) ↑	69.40(0.76) ↑	34.36(0.38) ↓
FedMut	98.70(0.11) ↓	89.14(0.09) ↑	82.50(0.77) ↑	56.17(0.72) ↑	64.92(0.37) ↑	69.10(1.00) ↑	69.70(1.07) ↑	35.28(0.78) ↑
FedACG	98.92(0.26) ↑	88.56(0.10) ↓	81.88(1.27) ↑	55.82(0.81) ↑	64.92(0.37) ↑	68.10(1.00) ↑	69.82(1.07) ↑	35.19(0.79) ↑
Ours	**99.12**(0.07) ↑	89.28(0.13) ↑	**85.37**(0.96) ↑	**57.73**(0.92) ↑	**66.47**(0.51) ↑	**70.35**(0.54) ↑	**71.12**(0.98) ↑	**36.85**(0.84) ↑

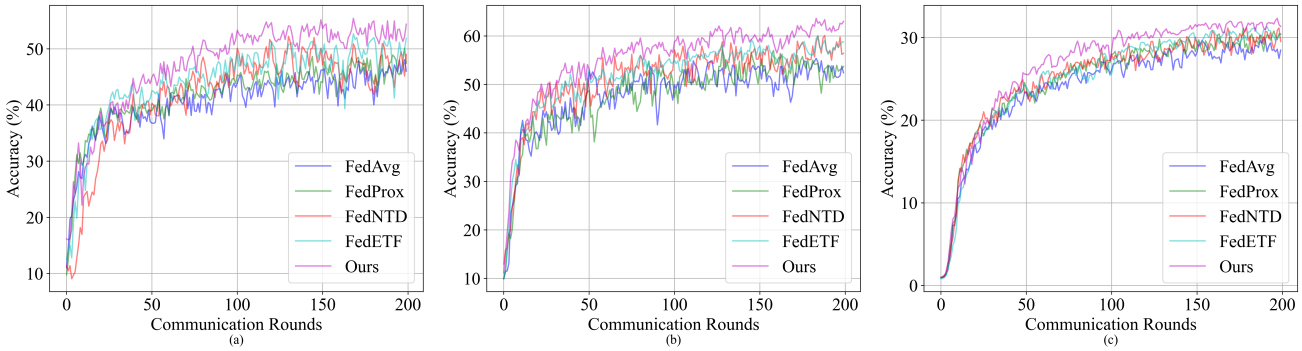

Fig. 5. Accuracy curves of global model tested using different methods. (a) CIFAR-10 with $dir = 0.1$. (b) CIFAR-10 with $dir = 0.3$. (c) CIFAR-100 with $dir = 0.2$.

method is larger than the baseline. In particular, when the CIFAR-10 dataset is partitioned with $dir = 0.1$, our method improves by 10.72% compared to the baseline. Our method achieves optimal accuracy in most cases compared to previous methods. Even if it is not optimal in some cases, our method is close to the state-of-the-art.

We also present the test accuracy curves during the training process for our method and some other methods in Fig. 5. Experiments are conducted on the CIFAR-10 and CIFAR-100 datasets, and it can be observed that our method demonstrates higher accuracy and stability.

D. Discussion

1) Model Architecture: We evaluate the performance of various algorithms on different network architectures using the CIFAR-10 dataset. The results are presented in Tab. III. It can be observed that as the network architecture deepens, the model's accuracy tends to increase. Additionally, our method seamlessly integrates well with these model architectures.

TABLE III

THE TEST ACCURACY (%) UNDER DIFFERENT NETWORKS.

Method	**CNN**	**DNN**	**ResNet-10**	**ResNet-18**	**VGG-19**
FedAvg	46.82	32.16	46.52	47.01	47.28
FedProx	48.37	33.43	48.71	50.90	49.56
FedFN	50.99	40.47	52.37	54.10	54.31
FedNTD	52.71	38.84	54.42	55.83	54.87
Ours	**53.78**	**44.83**	**56.34**	**56.81**	**57.78**

TABLE IV
EFFECT OF HYPERPARAMETERS β AND μ ON CIFAR-10.

β	0.15	0.5	1.5	5.0	15.0	30.0	50.0
Accuracy	18.23	25.40	38.69	51.97	**56.48**	54.62	52.92
μ	0.05	0.15	0.5	1.5	5.0	10.0	15.0
Accuracy	45.15	49.15	51.87	53.68	**56.96**	55.26	52.49
τ	0.05	0.1	0.3	1.0	3.0	10.0	30.0
Accuracy	52.7	54.7	56.25	**56.89**	55.46	52.28	48.56

2) Hyperparameter Analysis: Tab. IV presents the performance of our proposed method under different combinations of hyperparameters. Our method has four hyperparameters: α, which represents inter-class separability; β, which signifies intra-class compactness; μ, which denotes the distillation weight; and τ, which indicates the smoothness of soft labels. In our experiments, we default to $\alpha = 0.5$, $\beta = 15$, $\mu = 5$, and $\tau = 1$.

Our method is not sensitive to α. When the other hyperparameters are fixed, a higher β results in higher model accuracy. This suggests that during local model training, enhancing the ability of features to cluster around their prototypes enables the local model to more adequately learn knowledge from its own data. Additionally, we also test the impact of μ on the model. It should fall within a reasonable range, an excessively large μ can prevent the local model from learning from local data due to excessive influence from the global model. Meanwhile, when the τ is small, knowledge distillation will overly strengthen the restrictions imposed by the global model on the local model, thereby suppressing the learning ability of the local model. When the τ is large, the local model will have difficulty retaining the knowledge of the global model, making it more prone to drift.

VI. CONCLUSION

In this paper, we introduce a novel loss in FL, allowing the model to train the feature extractor and classifier separately. To ensure the effectiveness of this loss function in a FL setting, we emphasize the importance of capturing knowledge beyond the user distribution and utilize distillation techniques to prevent forgetting this crucial information. Experimental results demonstrate that the proposed method achieves state-of-the-art performance without requiring additional privacy information. In future work, we will focus on applying our method to larger-scale practical scenarios and achieving exceptional performance.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (62106217), the Yunnan Provincial Department of Science and Technology Basic Research Program (202201AU070112), and the Young Talent Program of Yunnan Province (C619300A116).

REFERENCES

[1] J. Pei, W. Liu, J. Li, L. Wang, and C. Liu, "A review of federated learning methods in heterogeneous scenarios," *IEEE Transactions on Consumer Electronics*, 2024.

[2] 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.

[3] G. Zheng, J. Wang, L.-C. Yu, and X. Zhang, "Instruction tuning with retrieval-based examples ranking for aspect-based sentiment analysis," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 4777–4788.

[4] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, p. 128019, 2024.

[5] L. Wang, J. Bian, L. Zhang, C. Chen, and J. Xu, "Taming cross-domain representation variance in federated prototype learning with heterogeneous data domains," *arXiv preprint arXiv:2403.09048*, 2024.

[6] Y. Dai, Z. Chen, J. Li, S. Heinecke, L. Sun, and R. Xu, "Tackling data heterogeneity in federated learning with class prototypes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, 2023, pp. 7314–7322.

[7] V. Papyan, X. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020.

[8] W. Liu, L. Yu, A. Weller, and B. Schölkopf, "Generalizing and decoupling neural collapse via hyperspherical uniformity gap," *arXiv preprint arXiv:2303.06484*, 2023.

[9] Z. Tang, Y. Zhang, S. Shi, X. Tian, T. Liu, B. Han, and X. Chu, "Fedimpro: Measuring and improving client update in federated learning," *arXiv preprint arXiv:2402.07011*, 2024.

[10] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, 2024.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[12] G. Kim, J. Kim, and B. Han, "Communication-efficient federated learning with accelerated client gradient," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 12 385–12 394.

[13] M. Hu, Y. Cao, A. Li, Z. Li, C. Liu, T. Li, M. Chen, and Y. Liu, "Fedmut: Generalized federated learning via stochastic mutation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 528–12 537.

[14] S. Kim, G. Lee, J. Oh, and S.-Y. Yun, "Fedfn: Feature normalization for alleviating data heterogeneity problem in federated learning," *arXiv preprint arXiv:2311.13267*, 2023.

[15] T. Tirer, H. Huang, and J. Niles-Weed, "Perturbation analysis of neural collapse," in *International Conference on Machine Learning*. PMLR, 2023, pp. 34 301–34 329.

[16] R. Wu and V. Papyan, "Linguistic collapse: Neural collapse in (large) language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 137 432–137 473, 2024.

[17] Z. Li, X. Shang, R. He, T. Lin, and C. Wu, "No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5319–5329.

[18] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[19] D. Yao, W. Pan, Y. Dai, Y. Wan, X. Ding, C. Yu, H. Jin, Z. Xu, and L. Sun, "Fedgkd: Toward heterogeneous federated learning via global knowledge distillation," *IEEE Transactions on Computers*, vol. 73, no. 1, pp. 3–17, 2023.

[20] H. Chen, H. Vikalo *et al.*, "The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation," *arXiv preprint arXiv:2301.08968*, 2023.

[21] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 461–38 474, 2022.