



No Fear of Classifier Biases: Neural Collapse Inspired Federated Learning with Synthetic and Fixed Classifier

Zexi Li¹ Xinyi Shang² Rui He¹ Tao Lin^{3*} Chao Wu^{1*}

¹Zhejiang University ²Xiamen University ³Westlake University

{zexi.li, ruihe, chao.wu}@zju.edu.cn shangxinyi@stu.xmu.edu.cn lintao@westlake.edu.cn

No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data

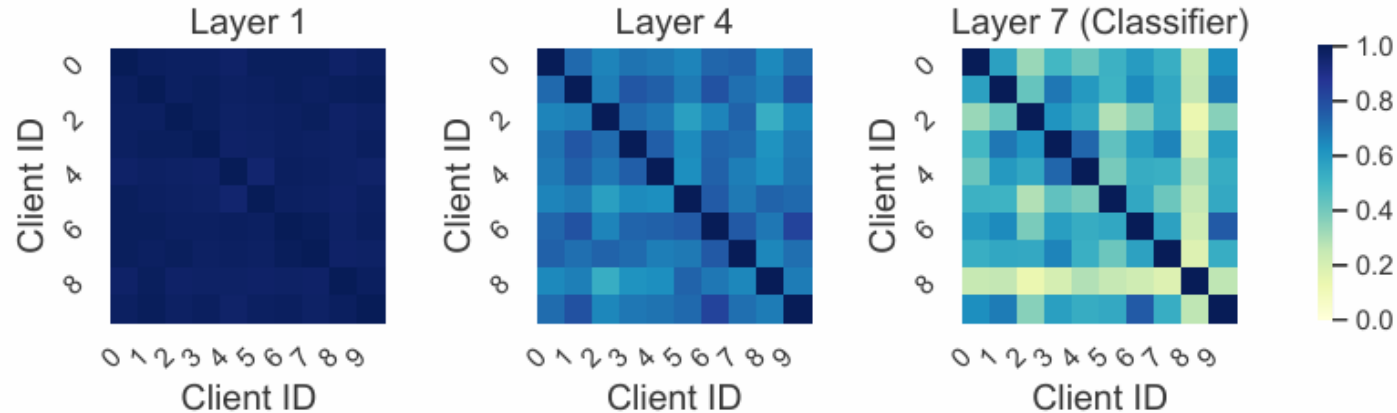


Figure 1: CKA similarities of three different layers of different 'client model-client model' pairs.

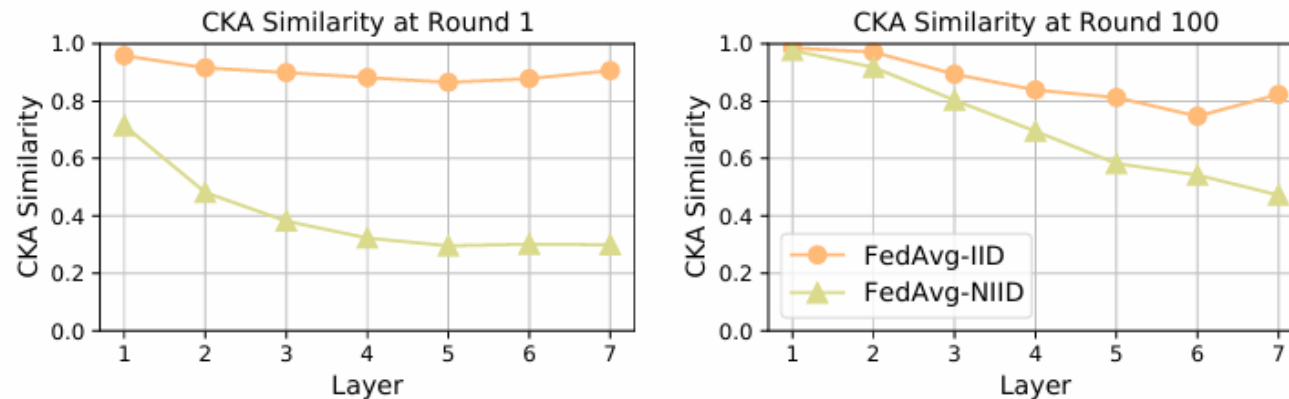


Figure 2: The means of the CKA similarities of different layers in different local models.

Classifier shows the lowest features similarities, among all the layers.
The low CKA similarities of the classifiers imply that the local classifiers change greatly to fit the local data distribution.

No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data

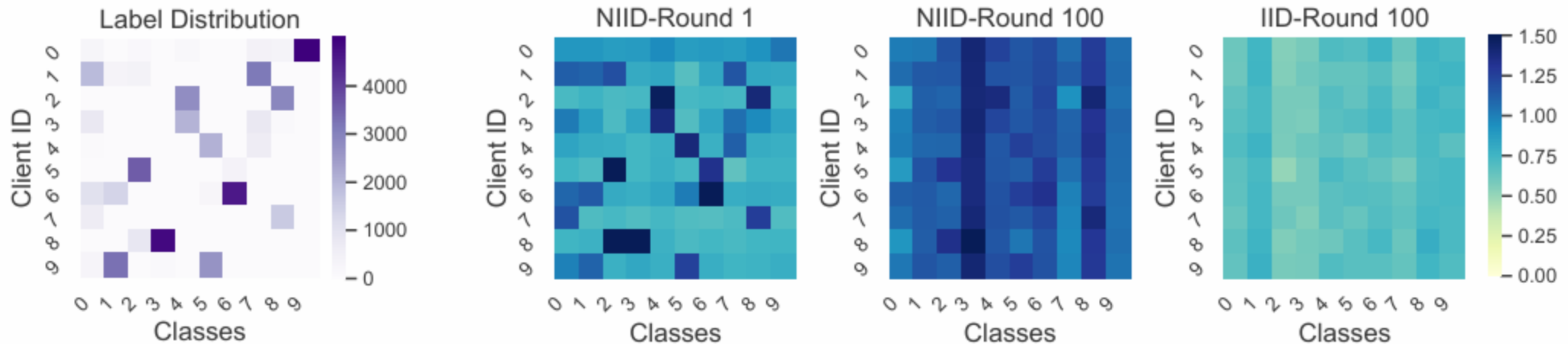


Figure 3: Label distribution of CIFAR-10 across clients (the first graph) and the classifier weight norm distribution across clients in different rounds and data partitions (the three graphs on the right).

The classifier weight norms would be biased to the class with more training samples at the initial training stage. At the end of the training, models trained on non-IID data suffer from a much heavier biased classifier than the models trained on IID data.

Existing methods for addressing the non-IID problem:

1. Mitigating Client Drift

Adjusting local objectives of clients to maintain some consistency between local and global models.

2. Modifying Aggregation Schemes

Enhancing the model aggregation mechanism at the server end.

3. Data Sharing

Introducing publicly available datasets or generating data to assist in constructing a more balanced data distribution at either the server or client end.

4. Personalized Federated Learning

Aiming to train personalized models for individual clients rather than a shared global model.

Neural Collapse:

The neural collapse was a phenomenon at the terminal phase of training on a balanced dataset, the feature prototypes and the classifier vectors will converge to a simplex ETF(Equiangular Tight Frame) where the vectors are normalized, and the pair-wise angles are maximized.

A collection of vectors $\mathbf{v}_i \in \mathbb{R}^d$, $d \geq C - 1$ is said to be a simplex equiangular tight frame if:

$$\mathbf{V} = \sqrt{\frac{C}{C-1}} \mathbf{U} \left(\mathbf{I}_C - \frac{1}{C} \mathbf{1}_C \mathbf{1}_C^T \right)$$

$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_C] \in \mathbb{R}^{d \times C}$, $\mathbf{U} \in \mathbb{R}^{d \times C}$ allows a rotation and satisfies $\mathbf{U}^T \mathbf{U} = \mathbf{I}_C$, \mathbf{I}_C is the identity matrix, and $\mathbf{1}_C$ is an all-ones vector.

Background



$$\mathbf{V} = \sqrt{\frac{C}{C-1}} \mathbf{U} \left(\mathbf{I}_C - \frac{1}{C} \mathbf{1}_C \mathbf{1}_C^T \right)$$

$\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_C] \in \mathbb{R}^{d \times C}$, $\mathbf{U} \in \mathbb{R}^{d \times C}$ allows a rotation and satisfies $\mathbf{U}^T \mathbf{U} = \mathbf{I}_C$, \mathbf{I}_C is the identity matrix, and $\mathbf{1}_C$ is an all-ones vector.

$$\mathbf{V} = \sqrt{\frac{C}{C-1}} (u_1, u_2, \dots, u_C) \begin{pmatrix} \frac{C-1}{C} & -\frac{1}{C} & \dots & -\frac{1}{C} \\ -\frac{1}{C} & \frac{C-1}{C} & & \\ \vdots & & \ddots & \\ -\frac{1}{C} & -\frac{1}{C} & & \frac{C-1}{C} \end{pmatrix}$$

$$\mathbf{V} = \sqrt{\frac{C}{C-1}} \left(\frac{C-1}{C} u_1 - \frac{1}{C} u_2 \dots - \frac{1}{C} u_C, -\frac{1}{C} u_1 + \frac{C-1}{C} u_2 \dots - \frac{1}{C} u_C, \dots \right)$$

$$u_1 \cdot v_1^T = \left(-2 \frac{C-1}{C^2} + \frac{C-2}{C^2} \right) \cdot \sqrt{\frac{C}{C-1}} \cdot \sqrt{\frac{C}{C-1}}$$

$$= -\frac{1}{C} \cdot \frac{C}{C-1}$$

$$= -\frac{1}{C-1}$$

Neural Collapse:

$$\mathbf{v}_i^T \mathbf{v}_j = \frac{C}{C-1} \delta_{i,j} - \frac{1}{C-1}, \forall i, j \in [C]$$

where $\delta_{i,j}$ equals to 1 when $i = j$ and 0 otherwise. The pair wise angle $-\frac{1}{C-1}$ is the maximal equiangular separation of C vectors in \mathbb{R}^d .

Three key properties of the neural collapse (NC) phenomenon:

NC1 (Features collapse to the class prototypes)

$$\Sigma_W^c \rightarrow \mathbf{0}, \text{ where } \Sigma_W^c := \frac{1}{n_c} \sum_{i=1}^{n_c} (\mathbf{h}_{c,i} - \mathbf{h}_c)(\mathbf{h}_{c,i} - \mathbf{h}_c)^T$$

NC2 (Prototypes collapse to simplex ETFs)

$$\tilde{\mathbf{h}}_c = (\mathbf{h}_c - \mathbf{h}_G) / \|\mathbf{h}_c - \mathbf{h}_G\| \quad \mathbf{h}_G = \sum_{c=1}^C \sum_{i=1}^{n_c} \mathbf{h}_{c,i}$$

NC3 (Classifiers collapse to the same simplex ETFs)

How data heterogeneity causes classifier biases in FL?

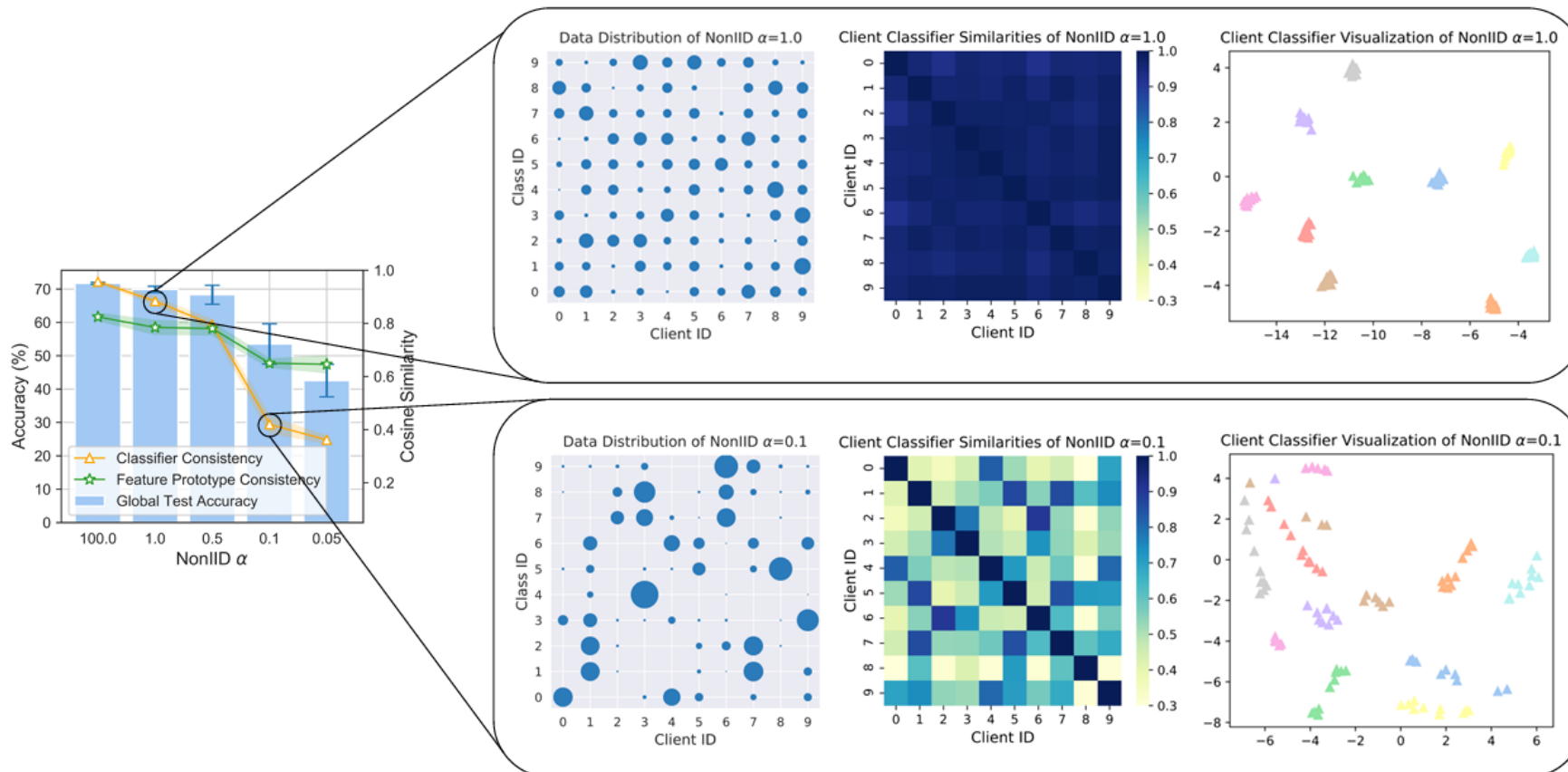
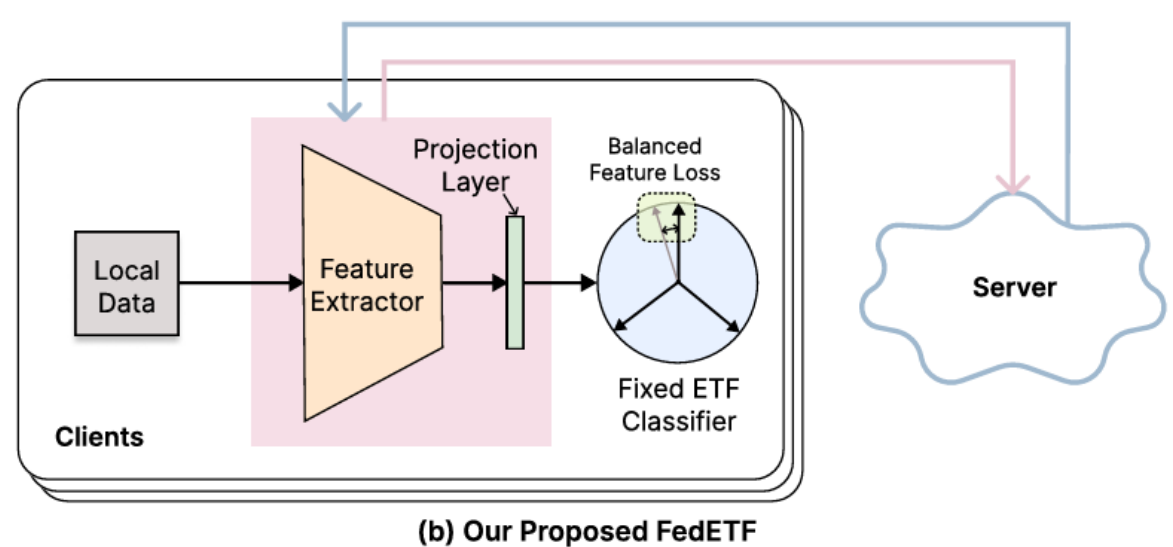
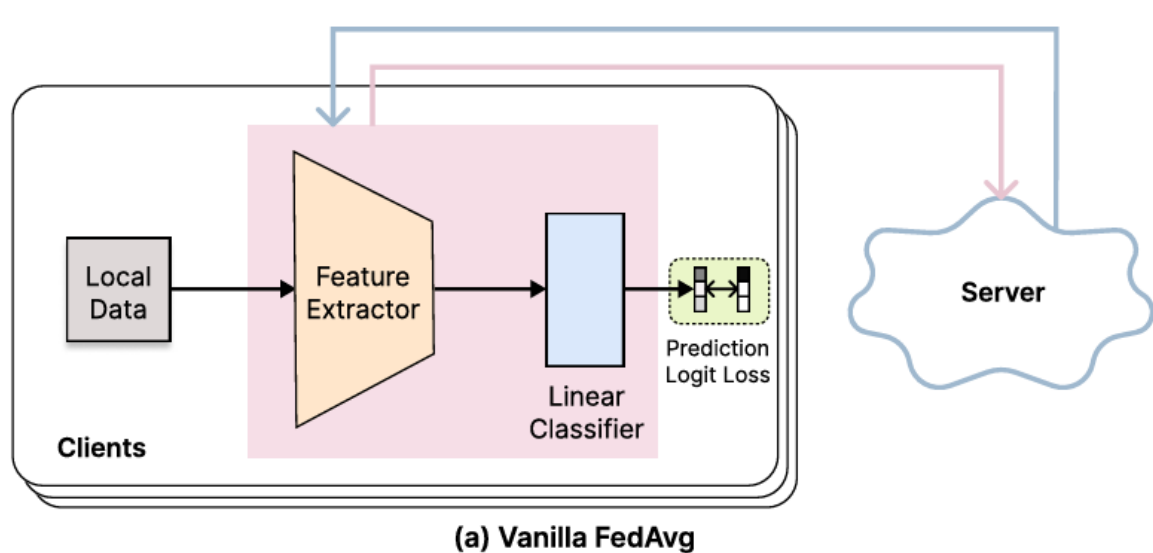


Figure 1. **How data heterogeneity causes classifier biases in FL.** Smaller α corresponds to higher Non-IID. Experiments are conducted on CIFAR-10 with vanilla FEDAVG. Columns from left to right: (1) Non-IID data results in poor generalization, biased classifiers, and misaligned features. (2) Clients' data distributions. (3) Clients with Non-IID data have smaller pair-wise classifier cosine similarities. (4) t-SNE visualization of clients' class-wise classifier vectors (represented by colors), which are more scattered in Non-IID data.

FEETF:

In FEETF, only the feature extractor and projection layer are learned and aggregated, and we adopt the same synthetic and fixed ETF classifier for all clients throughout the FL training process. Instead of prediction logit loss in vanilla FL, we use a novel balanced feature loss for the ETF classifier.

Network Architecture



ETF classifier initialization:

At the beginning of the FL training, we first randomly synthesize a simplex $\mathbf{V}_{ETF} \in \mathbb{R}^{d \times C}$, where d denotes the feature dimension of the ETF and C is the number of classes.

For each class's classifier vector \mathbf{v}_i , it requires $\|\mathbf{v}_i\|_2 = 1$.

Projection layer:

Given a data sample \mathbf{x} , we first use the feature extractor $f_{\mathbf{u}}$ to transform the data into the raw feature \mathbf{h} and then use a projection layer $g_{\mathbf{p}}$ to map this raw feature to the ETF feature space and normalize it into $\boldsymbol{\mu}$.

$$\boldsymbol{\mu} = \hat{\boldsymbol{\mu}} / \|\hat{\boldsymbol{\mu}}\|_2, \quad \hat{\boldsymbol{\mu}} = g(\mathbf{p}; \mathbf{h}), \quad \mathbf{h} = f(\mathbf{u}; \mathbf{x}),$$

- 1) If the last layer of the feature extractor is the non-linear activation, the raw feature \mathbf{h} will be sparse with zeros.
- 2) The raw features always have high dimensions, and high-dimensional vectors are more prone to be orthogonal.
- 3) The projection layer is helpful in the local finetuning stage for personalization.

Balanced feature loss with learnable temperature:

In neural-collapse-inspired imbalanced learning, it is found that when the ETF classifier is used, the gradients of cross entropy (CE) will be biased towards the head class. In FL, clients' local datasets are also class imbalanced due to data heterogeneity.

We define the model parameters in our FEETF as $\mathbf{w} = \{\mathbf{u}, \mathbf{p}, \beta\}$, which consists of the feature extractor, the projection layer, and the learnable temperature.

$$\ell^g(\mathbf{w}, \mathbf{V}_{ETF}; \mathbf{x}, y) = -\log \frac{n_{k,y}^{\gamma} \exp(\beta \cdot \mathbf{v}_y^T \boldsymbol{\mu})}{\sum_{c \in [C]} n_{k,c}^{\gamma} \exp(\beta \cdot \mathbf{v}_c^T \boldsymbol{\mu})}$$

where $n_{k,c}$ refers to the number of samples in class c of client k , β is the learnable temperature, $\boldsymbol{\mu}$ is the normalized feature.

Personalized Adaptation by Local Finetuning:

It consists of two parts: **local feature adaptation** and **classifier finetuning**.

In the local feature adaptation, we fix the projection layer and ETF classifier and finetune the feature extractor to let the feature extractor be more customized to the features of clients' local data.

In the classifier finetuning period, we finetune the ETF classifier and projection layer alternately for several iterations to make the classifier more biased to the local class distributions.

Personalized Adaptation by Local Finetuning:

The learned model parameters in the personalization stage are $\mathbf{w} = \{\mathbf{u}, \mathbf{p}, \beta, \bar{\mathbf{V}}_{ETF}\}$, and we split \mathbf{w} into the tuned parameters $\hat{\mathbf{w}}$ and the fixed parameters $\bar{\mathbf{w}}$. When finetune the feature extractor, $\hat{\mathbf{w}} = \{\mathbf{u}, \beta\}$, $\bar{\mathbf{w}} = \{\mathbf{p}, \mathbf{V}_{ETF}\}$; when finetune the ETF classifier, $\hat{\mathbf{w}} = \{\mathbf{V}_{ETF}, \beta\}$, $\bar{\mathbf{w}} = \{\mathbf{u}, \mathbf{p}\}$; when finetune the projection layer, $\hat{\mathbf{w}} = \{\mathbf{p}, \beta\}$, $\bar{\mathbf{w}} = \{\mathbf{u}, \mathbf{V}_{ETF}\}$. We use the vanilla CE loss without balanced softmax in each stage of finetuning.

$$\ell^p(\hat{\mathbf{w}}, \bar{\mathbf{w}}; \mathbf{x}, y) = -\log \frac{\exp(\beta \cdot \mathbf{v}_y^T \boldsymbol{\mu})}{\sum_{c \in [C]} \exp(\beta \cdot \mathbf{v}_c^T \boldsymbol{\mu})}.$$

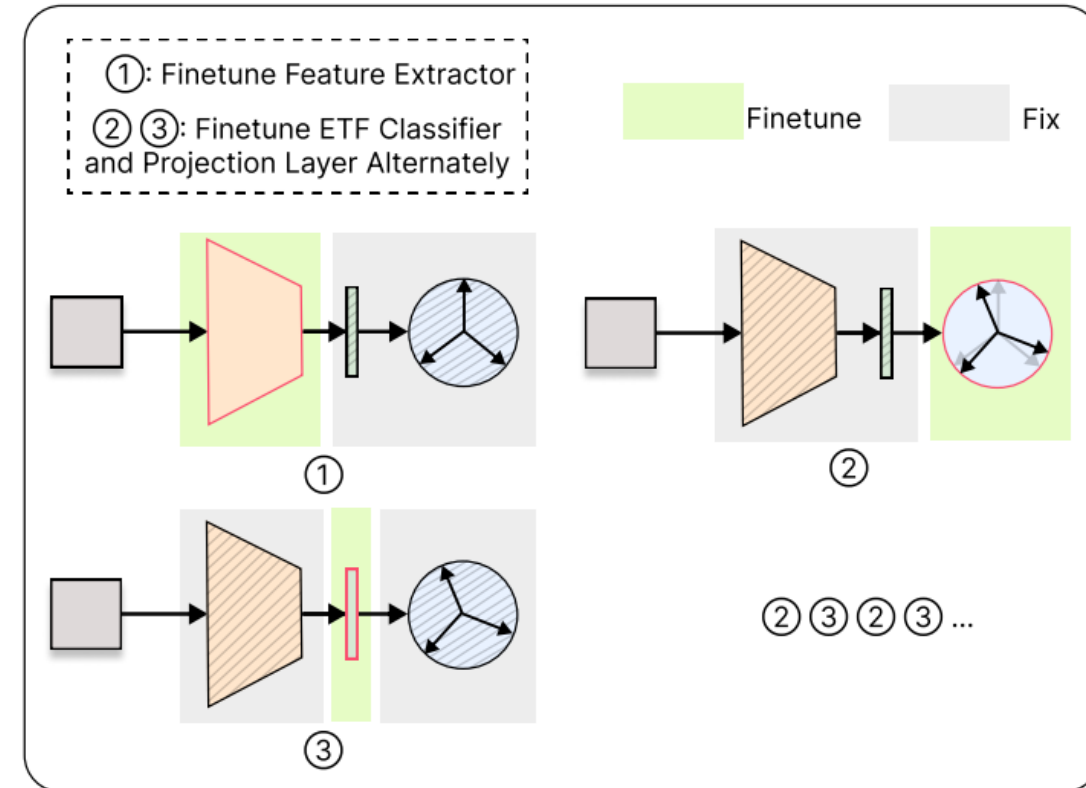


Table 1. Results in terms of generalization (General.) accuracy (%) of global models and personalization (Personal.) accuracy (%) of local models on three datasets under different heterogeneity. Best two methods in each setting are highlighted in **bold** fonts.

Dataset	CIFAR-10				CIFAR-100				Tiny-ImageNet			
NonIID (α)	0.1		0.05		0.1		0.05		0.1		0.05	
Methods/Metrics	General.	Personal.	General.	Personal.	General.	Personal.	General.	Personal.	General.	Personal.	General.	Personal.
FEDAVG [29]	52.76 \pm 6.08	83.85 \pm 0.89	44.48 \pm 6.19	89.80 \pm 0.39	24.77 \pm 1.19	49.93 \pm 1.17	22.53\pm0.40	58.85 \pm 0.33	28.93 \pm 0.52	40.81 \pm 0.35	24.88 \pm 0.34	46.90 \pm 0.44
FEDPROX [23]	46.59 \pm 3.04	82.08 \pm 0.27	40.95 \pm 5.75	87.69 \pm 2.85	23.33 \pm 1.72	46.44 \pm 1.64	19.12 \pm 0.77	57.01 \pm 2.17	25.93 \pm 0.27	31.90 \pm 1.91	23.06 \pm 0.68	32.43 \pm 0.65
FEDDYN [1]	36.35 \pm 5.33	85.39 \pm 0.77	23.90 \pm 1.40	88.72 \pm 1.59	25.53\pm2.39	51.79 \pm 2.12	20.71 \pm 2.83	61.77\pm0.32	26.42 \pm 0.56	45.84 \pm 0.34	23.63 \pm 1.55	52.27 \pm 1.06
DITTO [22]	52.76 \pm 6.08	79.81 \pm 1.89	44.48 \pm 6.19	85.17 \pm 3.47	24.77 \pm 1.19	38.06 \pm 1.26	22.53 \pm 0.40	50.18 \pm 1.22	28.93 \pm 0.52	33.00 \pm 1.01	24.88 \pm 0.34	40.31 \pm 0.12
FEDREP [4]	26.85 \pm 10.13	87.76\pm0.87	15.79 \pm 3.68	90.71 \pm 2.25	5.47 \pm 0.20	53.62\pm1.49	4.18 \pm 0.85	61.51 \pm 0.61	4.10 \pm 0.22	43.66 \pm 0.48	2.20 \pm 0.19	49.52 \pm 1.64
CCVR [28]	52.50 \pm 6.31	55.62 \pm 5.89	47.98 \pm 6.24	73.52 \pm 7.49	24.54 \pm 0.71	34.01 \pm 2.01	22.28 \pm 0.43	39.16 \pm 1.41	32.78\pm0.24	54.00\pm0.46	29.27\pm0.25	59.29\pm0.30
FEDPROTO [36]	-	83.34 \pm 0.71	-	88.21 \pm 1.77	-	43.31 \pm 0.70	-	54.87 \pm 0.52	-	40.74 \pm 0.87	-	48.05 \pm 0.82
FEDROD [3]	55.72\pm2.40	86.19 \pm 0.91	49.89\pm3.64	88.83 \pm 4.14	24.49 \pm 1.05	51.78 \pm 1.16	21.63 \pm 0.42	59.44 \pm 0.45	32.17 \pm 0.41	38.27 \pm 1.00	28.45 \pm 0.58	44.09 \pm 0.44
FEDNH [5]	55.37 \pm 4.48	85.98 \pm 0.15	47.96 \pm 2.59	91.06\pm3.13	24.67 \pm 0.68	52.09 \pm 0.78	21.95 \pm 0.85	62.71\pm0.22	17.51 \pm 0.62	36.53 \pm 0.29	14.00 \pm 0.17	41.80 \pm 1.78
Our FEDETF	59.56\pm1.84	87.89\pm1.19	56.08\pm3.44	92.62\pm0.54	26.24\pm1.78	52.86\pm1.53	24.17\pm0.54	60.68 \pm 0.91	33.49\pm0.82	55.82\pm0.60	29.15\pm1.03	62.36\pm0.13

- 1) Classical FL with Non-IID data: FEDAVG, FEDPROX, FEDDYN
- 2) Personalized FL: DITTO, FEDREP
- 3) FL methods most relevant to ours: CCVR, FEDPROTO, FEDROD, FEDNH

Communication Efficiency

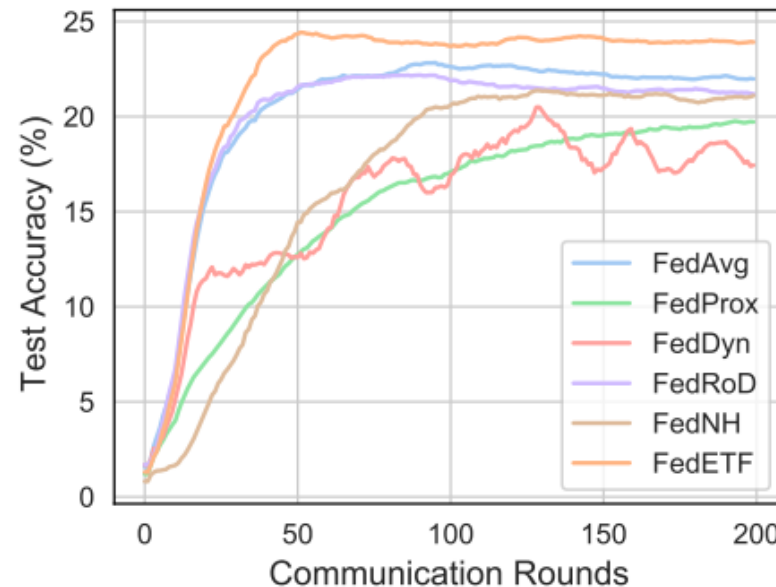
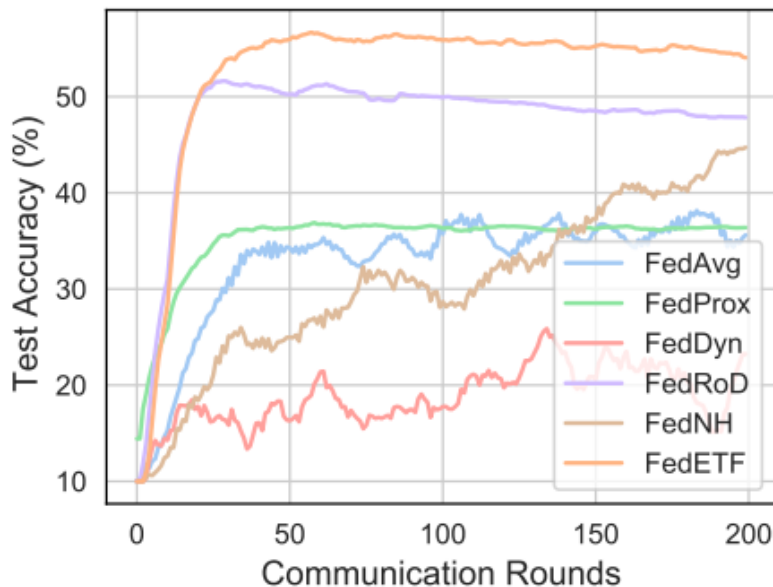


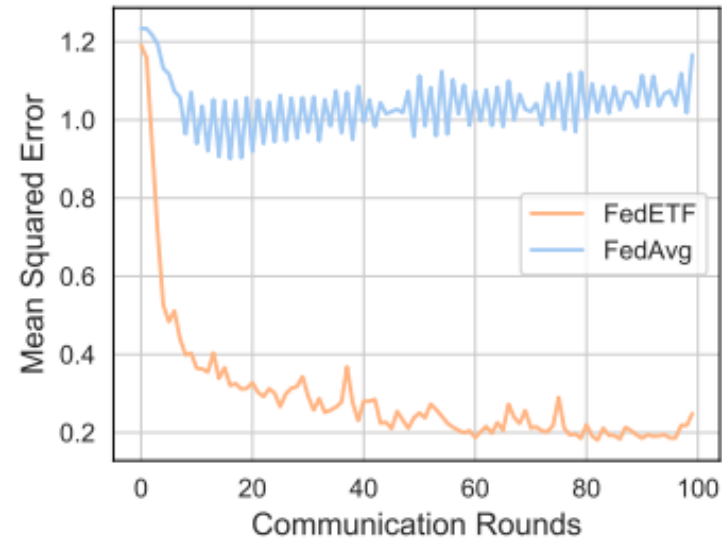
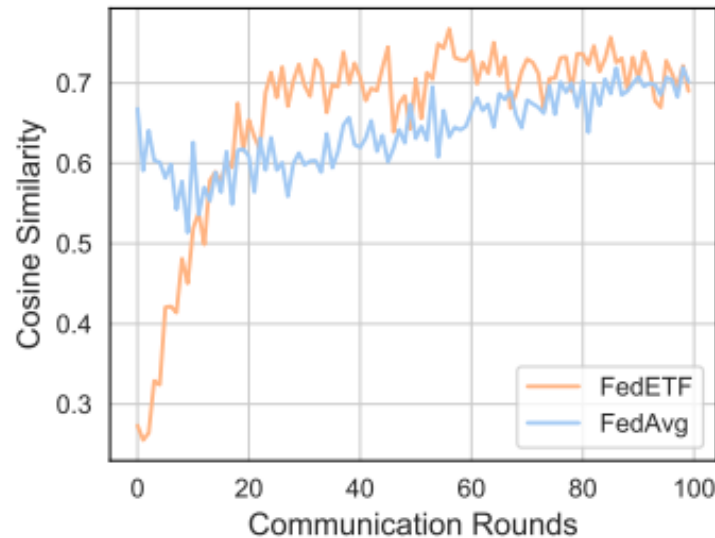
Table 2. Results (%) under various numbers of clients with partial client sampling. The dataset is CIFAR-10 with Non-IID $\alpha = 0.1$.

Number of Clients	50				100			
Sampling Rate	0.4		0.6		0.4		0.6	
Methods/Metrics	General.	Personal.	General.	Personal.	General.	Personal.	General.	Personal.
FEDAVG [29]	38.13 \pm 5.12	77.28 \pm 2.17	42.68 \pm 6.28	74.99 \pm 2.34	42.15 \pm 1.61	71.52 \pm 1.88	41.42 \pm 3.31	70.40 \pm 2.13
CCVR [28]	44.59 \pm 11.4	78.93\pm3.26	52.49 \pm 6.73	82.33\pm1.72	50.07 \pm 0.80	76.27\pm2.08	50.41 \pm 3.93	77.27\pm1.22
FEDROD [3]	55.84\pm3.96	76.60 \pm 0.13	53.04\pm2.54	74.42 \pm 1.99	52.62\pm1.68	71.27 \pm 0.69	52.34\pm0.11	72.41 \pm 0.74
FEDNH [5]	39.97 \pm 6.90	76.59 \pm 0.59	45.36 \pm 3.58	78.17 \pm 1.15	42.77 \pm 0.65	73.47 \pm 1.38	45.85 \pm 2.98	73.15 \pm 0.95
Our FEETF	58.05\pm4.63	85.82\pm0.86	58.75\pm1.72	85.05\pm0.87	56.67\pm0.88	83.47\pm0.45	55.96\pm0.23	83.38\pm0.72

Understanding FEETF

We first examine the feature alignment of local models. We compute class prototypes (feature mean of each class) in each client and calculate the cosine similarities of clients' class-wise prototypes, which is analogous to NC1.

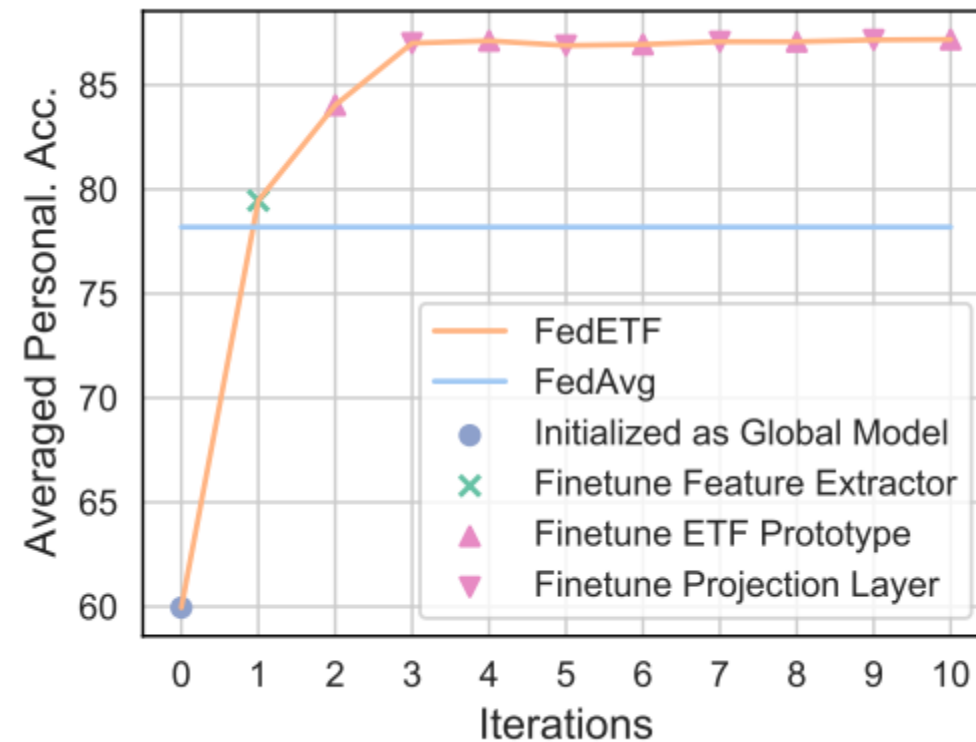
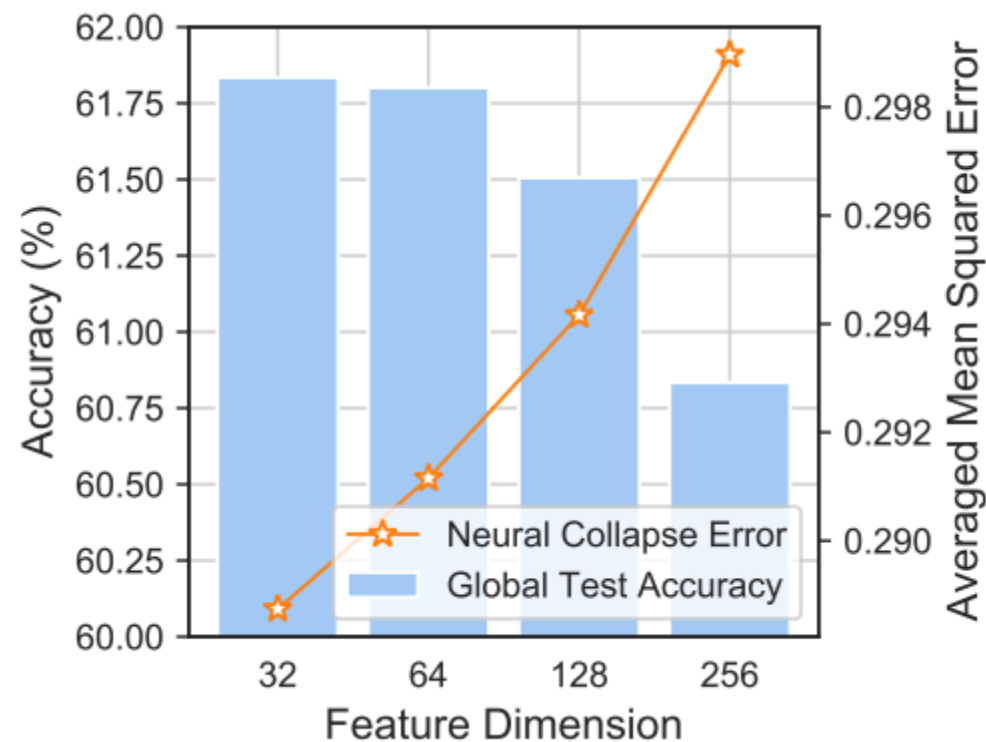
We compute the class prototypes of the global model and calculate the pair-wise cosine similarities of these prototypes in terms of NC2.



(a) Feature prototype consistency of clients' local models

(b) Neural collapse error of the aggregated global model

How feature dimension affects FEDETf



- (a) How feature dimension affects FEDETf's generalization and neural collapse.
- (b) How personalization is reached in each iteration of FEDETf's local finetuning.

In this paper, we fundamentally solve the classifier biases caused by data heterogeneity in FL by proposing a neural collapse-inspired solution. Specifically, we employ a simplex ETF as a fixed classifier for all clients during federated training, which allows them to learn unified and optimal feature representations. Afterward, we introduce a novel finetuning strategy to enable clients to have more personalized local models. Our method achieves the state-of-the-art performance regarding both generalization and personalization compared to strong baselines, as shown by extensive experimental results on CIFAR-10/100 and Tiny-ImageNet. Furthermore, we gained insights into understanding the effectiveness and applicability of our approach.

Thanks