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

Zexi Li\textsuperscript{1} Xinyi Shang\textsuperscript{2}\textsuperscript{*} Rui He\textsuperscript{1} Tao Lin\textsuperscript{3}\textsuperscript{*} Chao Wu\textsuperscript{1}\textsuperscript{*}
\textsuperscript{1}Zhejiang University \textsuperscript{2}Xiamen University \textsuperscript{3}Westlake University

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

1. Pseudo Codes of the Proposed Method

To better present our approach and demonstrate the workflow, we give the pseudo codes of FEDETF. The pseudo code of the federated learning (FL) training is shown in Algorithm 1 and the pseudo code of the personalized local finetuning is shown in Algorithm 2.

Algorithm 1: FEDETF FL Training

\textbf{Input:} Clients \{1, \ldots, K\}, communication round $T$, local epoch $E$, initial model $w^1 = \{u, p, \beta\}$, feature dimension $d$, balanced loss hyperparameter $\gamma$.

\textbf{Output:} Global model $w^g$.

Synthesize a simplex ETF $V_{ETF} \in \mathbb{R}^{d \times C}$ by Eq. (3) as the fixed classifier for all clients;

\textbf{for} $t = 1, \ldots, T$ \textbf{do}

\textbf{for} client $k = 1, \ldots, K$ \textbf{in parallel do}

$w_k^t \leftarrow w^t$;

\textbf{for} local epoch $e = 1, \ldots, E$ \textbf{do}

Obtain $L_k^p$ by Eq. (6, 7, 8);

$w_k^t \leftarrow w_k^t - \eta \nabla L_k^g(w_k^t)$;

\textbf{end}

The server updates $w^{t+1}$ by Eq. (2).

\textbf{end}

The final global model $w^g = w^T$.

Algorithm 2: FEDETF Personalized Finetuning

\textbf{Input:} Clients \{1, \ldots, K\}, iteration round $T_p$, epoch for each stage $E$, final global model $w^g = \{u, p, \beta, V_{ETF}\}$.

\textbf{Output:} Personalized local models $\{w_k^p\}_{k=1}^K$.

Assign the final global model $w^g$ as clients’ initial local models.

Finetune the feature extractor:

\textbf{for} client $k = 1, \ldots, K$ \textbf{in parallel do}

$w_k = \{u, \beta\}$, $\overline{w}_k = \{p, V_{ETF}\}$;

\textbf{for} local epoch $e = 1, \ldots, E$ \textbf{do}

Obtain $L_k^p$ by Eq. (9, 10, 11);

$\overline{w}_k \leftarrow \overline{w}_k - \eta \nabla L_k^p(\overline{w}_k)$;

\textbf{end}

\textbf{end}

\textbf{for} $t = 1, \ldots, T_p$ \textbf{do}

\textbf{for} client $k = 1, \ldots, K$ \textbf{in parallel do}

Finetune the ETF classifier:

$\hat{w}_k = \{V_{ETF}, \beta\}$, $\overline{w}_k = \{p, u\}$;

\textbf{for} local epoch $e = 1, \ldots, E$ \textbf{do}

Obtain $L_k^p$ by Eq. (9, 10, 11);

$\hat{w}_k \leftarrow \hat{w}_k - \eta \nabla L_k^p(\hat{w}_k)$;

\textbf{end}

Finetune the projection layer:

$\hat{w}_k = \{p, \beta\}$, $\hat{w}_k = \{V_{ETF}, u\}$;

\textbf{for} local epoch $e = 1, \ldots, E$ \textbf{do}

Obtain $L_k^p$ by Eq. (9, 10, 11);

$\hat{w}_k \leftarrow \hat{w}_k - \eta \nabla L_k^p(\hat{w}_k)$;

\textbf{end}

\textbf{end}

The personalized local models are $\{w_k^p = \hat{w}_k \cup \overline{w}_k\}_{k=1}^K$.

2. Implementation Details

Models and Data. Our model implementations of the ResNet series and the DenseNet are referred from the codes of [5]. The model implementation of the EfficientNet is referred from the official code in [9], and the implementation of the MobileNetv2 is referred from [8][4]. For the data, we use the Dirichlet-sampling-based data partition adopted in [6][1][2][7]. It considers a class-imbalanced data heterogeneity, controlled by hyperparameter $\alpha$, and smaller $\alpha$ refers to more Non-IID data of clients. When $\alpha < 1$, the data are considered to be rather Non-IID, which means that

\textsuperscript{*}Corresponding authors. \textsuperscript{†}Work was done during Xinyi’s visit to Westlake University.
most of the training samples of one class are likely assigned to a small portion of clients [1]. In our Dirichlet implementation, when $\alpha$ goes smaller, the number of samples in each client along with the class distribution of each client both become more heterogeneous, which is realistic in practical scenarios. We use the same Tiny-ImageNet dataset as in [2].

**Local learning rate and optimizer.** For CIFAR-10 the local learning rate (LR) $\eta = 0.04$, and for CIFAR-10 and Tiny-ImageNet, $\eta = 0.01$. For clients, we use SGD optimizer with momentum 0.9 and weight decay $5 \times 10^{-4}$. Following [3], we adopt a learning rate decaying scheduler, which decays the local LR by 0.99 in each round.

**Hyperparameters.** For FedETF, we set the feature dimension to the number of classes, i.e. $d = C$; the initial temperature $\beta = 1$; $\gamma = 1$. We set $\mu_{FedProx} = 0.001$ in FedProx and $\alpha_{FedDyn} = 0.01$ in FedDyn as suggested in their official implementations or papers. For Ditto, the learning setting of the personalized model is the same as the one of the global model. For FedRep, the epoch number of the classifier training and the epoch number of the feature extractor training are the same and are set as $E$. For FedRoD, we set $\gamma = 1$. For CCVR, the number of virtual features is 10 per class, and the number of classifier calibration training epochs is 100. For FedNH, the smoothing hyperparameter $\rho = 0.9$ as suggested in the paper [2].

**Randomness.** We set the same random seeds for all methods in the same setting. The random seed list is $\{7, 8, 9, 10\}$. For the extremely Non-IID settings when $\alpha = 0.05$, we use the random seeds that can ensure all clients can be assigned a proportion of training data (on the contrary, some random seeds will generate a data partition where particular clients have zero data samples).

**Environments.** All experiments are conducted in PyTorch with Quadro RTX 8000 GPUs.

### 3. Visualization

#### 3.1. Visualization of clients’ data distributions.

Here, we additionally visualize the clients’ data distributions mainly adopted in the main paper. In the main paper, we adopt $\alpha \in \{0.1, 0.05\}$ with 20 clients in Tables 1, 3, and 4. We visualize the data distributions in Figure 1. It shows that in both settings, the clients have extremely heterogeneous data distributions. Especially when $\alpha = 0.05$, all clients have some classes missing, and some clients have extremely rare data (e.g. client 1). We note that these settings are very realistic in practical FL scenarios.

**References**


