# Understanding and Mitigating Dimensional Collapse in Federated Learning

Yujun Shi Jian Liang Wenqing Zhang Chuhui Xue Vincent Y. F. Tan Song Bai

**Abstract**—Federated learning aims to train models collaboratively across different clients without sharing data for privacy considerations. However, one major challenge for this learning paradigm is the *data heterogeneity* problem, which refers to the discrepancies between the local data distributions among various clients. To tackle this problem, we first study how data heterogeneity affects the representations of the globally aggregated models. Interestingly, we find that heterogeneous data results in the global model suffering from severe *dimensional collapse*, in which representations tend to reside in a lower-dimensional space instead of the ambient space. This dimensional collapse phenomenon severely curtails the expressive power of models, leading to significant degradation in the performance. Next, via experiments, we make more observations and posit two reasons that result in this phenomenon: 1) dimensional collapse on local models; 2) the operation of global averaging on local model parameters. In addition, we theoretically analyze the gradient flow dynamics to shed light on how data heterogeneity result in dimensional collapse. To remedy this problem caused by the data heterogeneity, we propose FEDDECORR, a novel method that can effectively mitigate dimensional collapse in federated learning. Specifically, FEDDECORR applies a regularization term during local training that encourages different dimensions of representations to be uncorrelated. FEDDECORR, which is implementation-friendly and computationally-efficient, yields consistent improvements over various baselines on five standard benchmark datasets including CIFAR10, CIFAR100, TinyImageNet, Office-Caltech10, and DomainNet. Our code can be found at https://github.com/bytedance/FedDecorr.

**Index Terms**—Federated Learning, Representation Learning, Distribution Shift, Dimensional Collapse.

## 1 INTRODUCTION

WITH the rapid development of deep learning and the availability of large amounts of data, concerns regarding data privacy have been attracting increasingly more attention from industry and academia. To address this concern, [1] propose *Federated Learning*—a decentralized training paradigm enabling collaborative training across different clients without sharing data.

One major challenge in federated learning is the potential discrepancies in the distributions of local training data among clients, which is known as the *data heterogeneity* problem. In particular, analyses in this paper focus on the heterogeneity of *label distributions* (see Fig. 1(a) for an example). Such discrepancies can result in drastic disagreements between the local optima of the clients and the desired global optimum, which may lead to severe performance degradation of the global model. Previous works attempting to tackle this challenge mainly focus on the model parameters, either during local training [2], [3] or global aggregation [4]. However, these methods usually result in an excessive computation burden or high communication costs [5] because deep neural networks are typically heavily over-parameterized. In contrast, in this work, we focus on the representation space of the model and study the impact of data heterogeneity.

- *Manuscript received on Mar. 20 2023, revised on Sept. 11 2023*
- *Y. Shi and V. Y. F. Tan are with National University of Singapore (shi.yujun@u.nus.edu and vtan@nus.edu.sg).*
- *W. Zhang, C. Xue and S. Bai are with Bytedance Inc (wenqingzhang@bytedance.com, xuec0003@e.ntu.edu.sg, and songbai.site@gmail.com).*
- *J. Liang is with CRIPAC & MAIS, Institute of Automation, Chinese Academy of Sciences (liangjian92@gmail.com).*
- *Work done when Y. Shi was interning in Bytedance.*

To commence, we study how heterogeneous data affects the global model in federated learning in Sec. 3.1. Specifically, we compare representations produced by global models trained under different degrees of data heterogeneity. Since the singular values of the covariance matrix provide a comprehensive characterization of the distribution of high-dimensional embeddings, we use it to study the representations output by each global model. Interestingly, we find that as the degree of data heterogeneity increases, more singular values tend to evolve towards zero. This observation suggests that stronger data heterogeneity causes the trained global model to suffer from more severe *dimensional collapse*, whereby representations are biased towards residing in a lower-dimensional space (or manifold). A graphical illustration of how heterogeneous training data affect output representations is shown in Fig. 1(b-c). Our observations suggest that dimensional collapse might be one of the key reasons why federated learning methods struggle under data heterogeneity. Essentially, dimensional collapse is a form of *oversimplification* in terms of the model, where the representation space is not being fully utilized to discriminate diverse data of different classes.

Based on the observations made on the global model, we continue to explore the reasons for this phenomenon. To start with, since the global model is a result of the aggregation of local models, we conjecture that one reason behind dimensional collapse of the global model is the dimensional collapse of local models. To validate this conjecture, we visualize the local models in terms of the singular values of the representation covariance matrices in Sec. 3.2. We observe similar dimensional collapse phenomenon as in global models.

Next in Sec. 3.3, we investigate whether the operation

of global averaging itself will cause dimensional collapse. Specifically, we compare the representations produced by global and local models in terms of the singular values of their representation covariance matrices. Surprisingly, we find the singular values of global model are consistently lower than that of local models. This suggests that the global averaging operation, which averages parameters of local models into one global model, leads to dimensional collapse on the global model comparing to the original local models.

To further develop a more rigorous understanding on the dimensional collapse, we analyze the gradient flow dynamics of local training in Sec. 3.4. Interestingly, we show theoretically that heterogeneous data drive the weight matrices of the local models to be biased to being low-rank, which further results in representation dimensional collapse.

Inspired by the observations and analyses above, we develop a method to mitigate dimensional collapse during local training in Sec. 4. In particular, we propose a novel federated learning method called FEDDECORR. FEDDECORR adds a regularization term during local training to encourage the Frobenius norm of the correlation matrix of representations to be small. We show theoretically and empirically that this proposed regularization term can effectively mitigate dimensional collapse (see Fig. 1(d) for example). Next, in Sec. 5, through extensive experiments on standard benchmark datasets including CIFAR10, CIFAR100, TinyImageNet, Office-Caltech10, DomainNet, we show that FEDDECORR consistently improves over baseline federated learning methods. In addition, we find that FEDDECORR yields more dramatic improvements in more challenging federated learning setups such as stronger heterogeneity or a larger number of clients. Lastly, FEDDECORR has extremely low computation overhead and can be built on top of any existing federated learning baseline method, which makes it widely applicable.

Our contributions are summarized as follows. First, we discover through experiments that stronger data heterogeneity in federated learning leads to greater dimensional collapse for the global model. Second, we discover two underlying reasons behind dimensional collapse of the global model, namely dimensional collapse of the local models and the operation of global averaging on local model parameters. Third, we develop a theoretical understanding of the dynamics behind our empirical discovery that connects data heterogeneity and dimensional collapse. Fourth, based on the motivation of mitigating dimensional collapse, we propose a novel method called FEDDECORR, which yields consistent improvements under various data heterogeneity settings while being implementation-friendly and computationally-efficient.

This work extends our previous conference work presented in ICLR2023: Towards Understanding and Mitigating Dimensional Collapse in Federated Learning. Comparing to the conference version, we make the following substantial improvements:

- We discover a new reason that can lead to dimensional collapse on the global model, namely the operation of averaging local model parameters. We also empirically show that our proposed method FEDDECORR can alleviate the adverse outcomes due to dimensional collapse from this newly uncovered reason.
- Different from the conference version, we empirically show that FEDDECORR can even be employed beyond label heterogeneity and improve the classification performance in federated learning under *domain heterogeneity*.
- We add significantly more experiments to better validate our proposed method FEDDECORR. This includes: 1) experiments demonstrating the effectiveness of FEDDECORR on a new type of label heterogeneity partition: pathological non-iid partition; 2) comparisons in terms of computational efficiencies between FEDDECORR and previous methods; 3) comparisons between FEDDECORR and other decorrelation methods; 4) experiments on different neural network architectures.
- We have polished writing of the abstract, introduction, methodology, experiment sections and included more thorough discussion on related works.

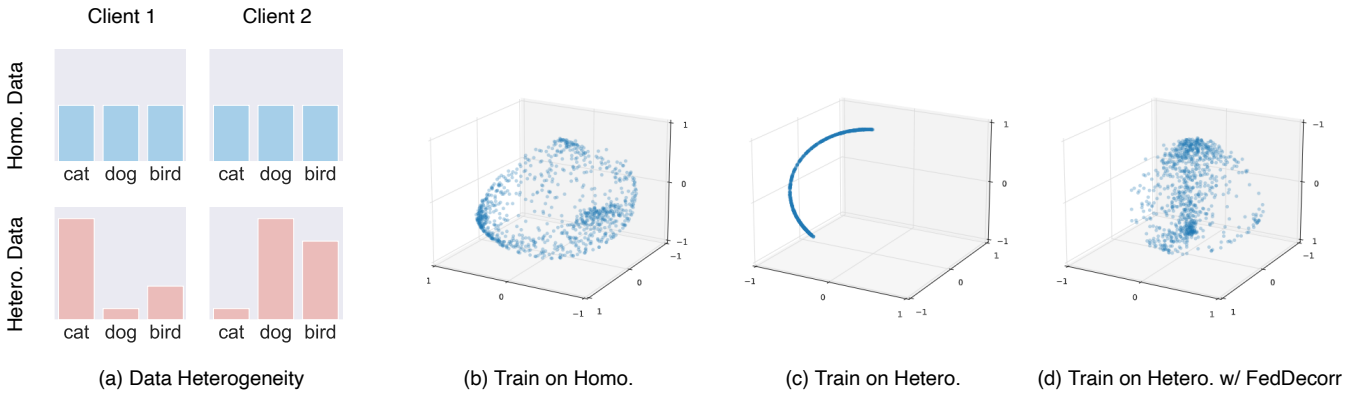
## 2 RELATED WORKS

### 2.1 Federated Learning

Federated Learning is a decentralized learning paradigm proposed originally in [1]. It strives to train a effective machine learning model without collecting local data of each client. As a first step to solving this problem, [1] proposed FedAvg, which adopts a simple averaging scheme to aggregate local models into the global model. However, under data heterogeneity, FedAvg suffers from unstable and slow convergence, resulting in performance degradation. To tackle this challenge, previous works either improve local training [2], [3], [6]–[12] or global aggregation [4], [13]–[15], [15]–[18]. Most of these methods focus on the model parameter space, which may result in high computation or communication cost due to deep neural networks being over-parameterized. For example, [2] requires computing regularization terms based on all model parameters, which can be very computational intensive; [3] requires exchanging extra information between the server and clients that is of the same size as the model parameters. Different from these works, [19], [20] propose to adjust the model output logit during local training to counter the label heterogeneity, which is much more efficient. However, one major problem of these methods is that their application is restricted to the scenario of label heterogeneity. [6] focuses on model representations and uses a contrastive loss to maximize agreements between representations of local models and the global model. However, one drawback of [6] is that it requires additional forward passes during training, which almost doubles the training cost. In this work, based on our study of how data heterogeneity affects model representations, we propose an effective yet highly efficient method to handle heterogeneous data.

Another research trend is in personalized federated learning [21]–[28], which aims to train personalized local models for each client. In this work, we apply FEDDECORR on a personalized federated learning baseline, namely FedBN [28], under the domain heterogeneity setting.

Besides the parameters-sharing-based federated learning settings mentioned before, another setting of federated




(a) Data Heterogeneity (b) Train on Homo. (c) Train on Hetero. (d) Train on Hetero. w/ FedDecorr

Fig. 1: **(a)** illustrates data heterogeneity in terms of number of samples per class. **(b)**, **(c)**, **(d)** show representations (normalized to the unit sphere) of global models trained under homogeneous data, heterogeneous data, and heterogeneous data with FEDDECORR, respectively. Only **(c)** suffers dimensional collapse. **(b)**, **(c)**, **(d)** are produced with ResNet20 on CIFAR10. Best viewed in color.

learning is vertical federated learning [29]–[32]. Under this setting, different *features* instead of different *data samples* are being split across different clients. Moreover, the information being exchanged between the server and the clients are the intermediate activation of neural networks. In this work, however, we only focus on the more popular horizontal federated learning setting.

### 2.2 Model Fusion

Another direction related to Federated Learning is Model Fusion, which also study how to better merge different models so that the performance of the merged model does not decrease. However, there are two major distinctions between these two similar topics: 1) model fusion includes studying how to merge two models independently trained on the exact same dataset [33]–[37] while federated learning only focuses fusing models trained on different datasets; 2) training data are usually strictly inaccessible in federated learning, but it is not the case for model fusion [37]. [38] first investigate how to align neural network output units with matching algorithms. Later, [33]–[35], [37] further explore the idea of permuting neurons for better fusing different models. [36] directly averaging weights of different fine-tuned models and achieves the state-of-the-art performance on ImageNet.

### 2.3 Dimensional Collapse

Dimensional collapse is the phenomenon of model output representations being biased towards residing in a lower dimensional space (or manifold) instead of the full ambient space. This phenomenon has been studied in metric learning [39], self-supervised learning (SSL) [40]–[42], and class incremental learning (CIL) [43]. Specifically, [39] investigates the relation between dimensional collapse and generalization capability of deep metric learning models; [40]–[42] study how to prevent dimensional collapse in SSL; [43] proposes to handle dimensional collapse at the initial phase to improve CIL performance. In this work, however, we focus on federated learning and discover two distinct factors that can cause representations to suffer dimensional collapse: strong local data heterogeneity and the operation of averaging model parameters of local clients. To the best of our knowledge, this work is the first to discover and analyze dimensional collapse of representations in federated learning.

### 2.4 Gradient Flow Dynamics

[44], [45] introduce the gradient flow dynamics framework to analyze the dynamics of multi-layer linear neural networks under the $\ell_2$-loss and find deeper neural networks that are biased towards low-rank solution during optimization. Following their works, [40] finds two factors that cause dimensional collapse in contrastive self-supervised learning, namely, strong data augmentation and implicit regularization from depth. [46] use gradient flow dynamics to analyze how non-contrastive self-supervised learning methods can circumvent mode collapse. Different from previous works, we focus on federated learning with the cross-entropy loss. More importantly, our analysis focuses on understanding dimensional collapse caused by data heterogeneity in federated learning instead of the depth of neural networks or strong data augmentations.

### 2.5 Feature Decorrelation

Feature decorrelation had been used for different purposes, such as preventing mode collapse in self-supervised learning [41], [42], [47], [48], boosting generalization [49]–[51], and improving class incremental learning [43]. Among these works, [47]–[49], [51] apply regularization terms on the representations during training to reduce correlation among different dimension of model output representations, while [41], [42], [50] apply whitening to explicitly enforce decorrelated representations. In this work, we apply feature decorrelation to counter the undesired dimensional collapse caused by data heterogeneity in federated learning. In addition, we adopt the regularization-based decorrelation approach instead of whitening. This is because the computation procedure of whitening is much more complicated than that of decorrelation regularization.

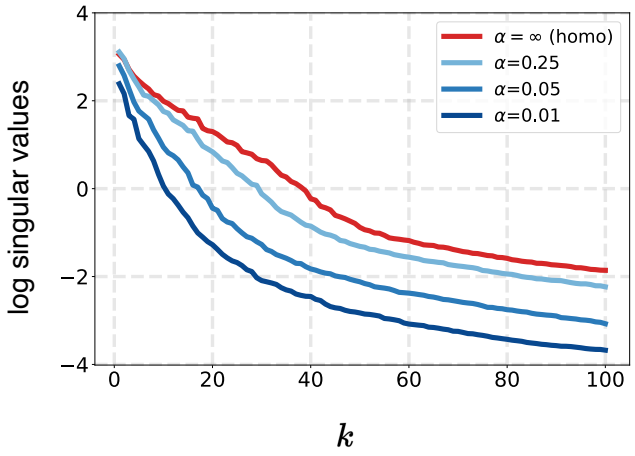



Fig. 2: Data heterogeneity causes dimensional collapse on **global models**. We plot the singular values of the covariance matrix of representations in descending order. The $x$-axis ($k$) is the index of singular values and the $y$-axis is the logarithm of the singular values. This figure shows that stronger heterogeneity results in a more drastic decrease in the singular values for an arbitrary $k$, indicating stronger dimensional collapse.

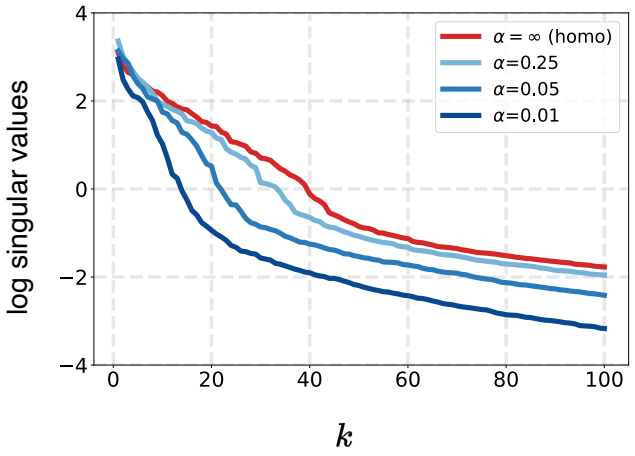



Fig. 3: Data heterogeneity causes dimensional collapse on **local models**. We plot the singular values of the covariance matrix of representations in descending order. The $x$-axis ($k$) is the index of singular values and the $y$-axis is the logarithm of the singular values. This figure also shows that stronger heterogeneity results in more drastic decrease in the singular values for an arbitrary $k$, indicating stronger dimensional collapse similarly to Fig. 2.

## 3 Dimensional Collapse from Data Heterogeneity

In this section, we first empirically visualize and compare representations of global models trained under different degrees of data heterogeneity in Sec. 3.1. Next, we provide empirical analyses in Sec. 3.2 and Sec. 3.3 and uncover two distinct reasons behind dimensional collapse on the global models. Finally, to theoretically understand our observations, we analyze the gradient flow dynamics in Sec. 3.4.

### 3.1 Empirical Observations on the Global Model

We first empirically demonstrate that stronger data heterogeneity causes more severe dimensional collapse on the global model. Specifically, we first separate the training samples of CIFAR100 into 10 splits, each corresponding to the local data of one client. To simulate data heterogeneity among clients as in previous works [6], [15], [52], we sample a probability vector $\mathbf{p}_c = (p_{c,1}, p_{c,2}, \ldots, p_{c,K}) \sim \mathrm{Dir}_K(\alpha)$ and allocate a $p_{c,k}$ proportion of instances of class $c \in [C] = \{1, 2, \ldots, C\}$ to client $k \in [K]$, where $\mathrm{Dir}_K(\alpha)$ is the Dirichlet distribution with $K$ categories and $\alpha$ is the concentration parameter. A smaller $\alpha$ implies stronger data heterogeneity ($\alpha = \infty$ corresponds to the homogeneous setting). We let $\alpha \in \{0.01, 0.05, 0.25, \infty\}$.

For each of the settings generated by different $\alpha$'s, we apply FedAvg [1] to train a MobileNetV2 [53] with CIFAR100 (observations on other federated learning methods, model architectures, or datasets are similar and are provided in the Supplementary Material Sec. C). Next, for each of the four trained global models, we compute the covariance matrix $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}_i - \bar{\mathbf{z}})(\mathbf{z}_i - \bar{\mathbf{z}})^\top$ of the representations over the $N$ test data points in CIFAR100. Here $\mathbf{z}_i$ is the $i$-th test data point and $\bar{\mathbf{z}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{z}_i$ is their average.

Finally, we apply the singular value decomposition (SVD) on each of the covariance matrices and visualize the top 100 singular values in Fig. 2. If we define a small value $\tau$ as the threshold for a singular value to be *significant* (e.g., $\log\tau = -2$), we observe that for the homogeneous setting, all the singular values are significant, i.e., they surpass $\tau$. However, as $\alpha$ decreases, the number of singular values exceeding $\tau$ monotonically decreases. This implies that with stronger heterogeneity among local training data, the representation vectors produced by the trained global model tend to reside in a lower-dimensional space, corresponding to more severe dimensional collapse.

### 3.2 Empirical Observations on Local Models

Since the global model is obtained by aggregating locally trained models on each client, we posit that one reason behind the dimensional collapse on the global model is the dimensional collapse of local models. To further validate this conjecture, we continue to study whether increasing data heterogeneity will also lead to more severe dimensional collapse on locally trained models.

Specifically, for different $\alpha$'s, we visualize the locally trained model of one client (visualizations on local models of other clients are similar and are provided in the Supplementary Materials Sec. D). Following the same procedure as in Sec. 3.1, we plot the singular values of covariance matrices of representations produced by the local models. We observe from Fig. 3 that locally trained models demonstrate the same trend as the global models—namely, that the presence of stronger data heterogeneity causes more severe dimensional collapse. These experiments corroborate that the global model inherit the adverse dimensional collapse phenomenon from the local models.

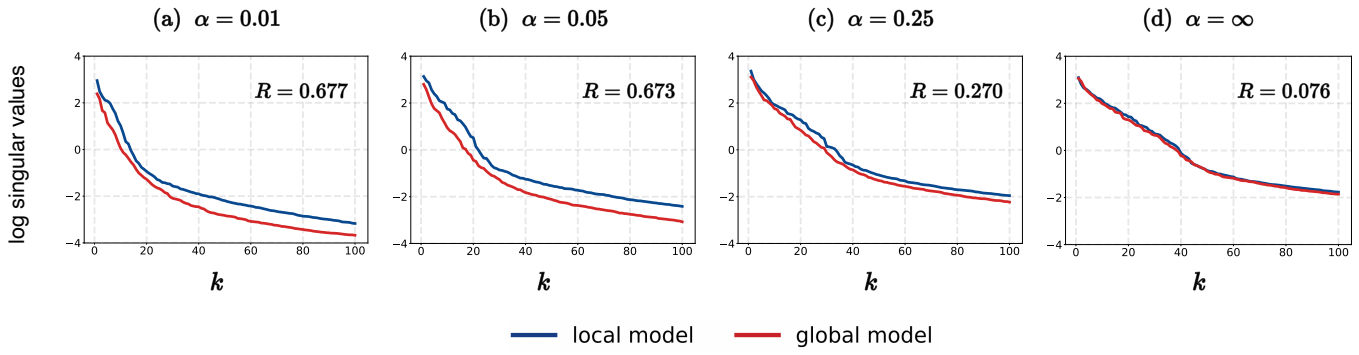



Fig. 4: **Global averaging causes dimensional collapse.** We compare representations between local and global models under different degrees of data heterogeneity. The $x$-axis ($k$) represents the indices of the singular values and the $y$-axis is the logarithm of the singular values. $R$ defined in Eqn. (1) are computed and are shown on top-right corner. In heterogeneous scenarios such as (a-c), averaging local models into a global model results in a drop in the singular values for every $k$.

### 3.3 Comparisons Between Local and Global Models

To further understand why the global model suffers from dimensional collapse under data heterogeneity, we provide comparisons between representations produced by local and global models. Specifically, under different $\alpha$'s, we plot the singular values of the covariance matrices of local and global model's output representations in Fig. 4. We also define the following metric $R$ to quantitatively compare the curves of local and global models:

$$R = \frac{1}{K}\sum_{k=1}^{K} \log \frac{\lambda_k^{(\mathrm{l})}}{\lambda_k^{(\mathrm{g})}}, \tag{1}$$

where $K$ is the total number of singular values, $\lambda_k^{(\mathrm{l})}$ and $\lambda_k^{(\mathrm{g})}$ are the $k$-th singular value of local and global model curve, respectively. Smaller $R$ indicates a smaller gap between the two curves.

From the results, we observe that for the homogeneous case (Fig. 4(d)), the curves for local and global models are almost the same, indicating that global averaging will not incur dimensional collapse on global model in this case.

However, as the heterogeneity becomes more severe (Fig. 4(a-c)), the curve of the global model is always below that of the local model, and the gap between them increase. Therefore, we have demonstrated that under data heterogeneity, the operation of global averaging on local models will also lead to dimensional collapse on the global model.

Although we have identified two distinct reasons for the dimensional collapse of the global model, the dimensional collapse on the local models is the more prominent one according to our visualizations.

### 3.4 A Theoretical Explanation for Dimensional Collapse

Based on the previous empirical observations, we now develop a theoretical understanding to explain why heterogeneous training data causes dimensional collapse for the learned representations.

Since we have observed that the dimensional collapse on local models is the more prominent factor that leads to global model dimensional collapse, we focus our attention on local models in this section. Without loss of generality, we study local training of one arbitrary client. Specifically, we first analyze the gradient flow dynamics of the model weights during the local training. This analysis shows how heterogeneous local training data drives the model weights towards being low-rank, which leads to dimensional collapse for the representations.

#### 3.4.1 Setups and Notations

We denote the number of training samples as $N$, the dimension of input data as $d_{\text{in}}$, and total number of classes as $C$. The $i$-th sample is denoted as $X_i \in \mathbb{R}^{d_{\text{in}}}$, and its corresponding one-hot encoded label is $\mathbf{y}_i \in \mathbb{R}^C$. The collection of all $N$ training samples is denoted as $X = [X_1, X_2 \ldots, X_N] \in \mathbb{R}^{d_{\text{in}} \times N}$, and the $N$ one-hot encoded training labels are denoted as $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbb{R}^{C \times N}$.

For simplicity in exposition, we follow [44], [45] and [40] and analyze linear neural networks (without nonlinear activation layers). We consider an $(L+1)$-layer (where $L \geq 1$) linear neural network trained using the cross entropy loss under gradient flow (i.e., gradient descent with an infinitesimally small learning rate). The weight matrix of the $i$-th layer ($i \in [L+1]$) at the optimization time step $t$ is denoted as $W_i(t)$. The dynamics can be expressed as

$$\dot{W}_i(t) = -\frac{\partial}{\partial W_i}\ell(W_1(t), \ldots, W_{L+1}(t)), \tag{2}$$

where $\ell$ denotes the cross-entropy loss.

In addition, at the optimization time step $t$ and given the input data $X_i$, we denote $\mathbf{z}_i(t) \in \mathbb{R}^d$ as the output representation vector ($d$ being the dimension of the representations) and $\gamma_i(t) \in \mathbb{R}^C$ as the output softmax probability vector. We have

$$\begin{aligned}\gamma_i(t) &= \mathrm{softmax}(W_{L+1}(t)\mathbf{z}_i(t)) \\ &= \mathrm{softmax}(W_{L+1}(t)W_L(t)\ldots W_1(t)X_i).\end{aligned} \tag{3}$$

We define $\mu_c = \frac{N_c}{N}$, where $N_c$ is number of data samples belonging to class $c$. We denote $\mathbf{e}_c$ as the $C$-dimensional one-hot vector where only the $c$-th entry is 1 (and the others are 0). In addition, let $\bar{\gamma}_c(t) = \frac{1}{N_c}\sum_{i=1}^{N}\gamma_i(t)\mathbb{1}\{\mathbf{y}_i = \mathbf{e}_c\}$ and $\bar{X}_c = \frac{1}{N_c}\sum_{i=1}^{N} X_i \mathbb{1}\{\mathbf{y}_i = \mathbf{e}_c\}$.

#### 3.4.2 Analysis on Gradient Flow Dynamics

Since our goal is to analyze model representations $\mathbf{z}_i(t)$, we focus on weight matrices that directly produce representations (i.e., the first $L$ layers). We denote the product of the weight matrices of the first $L$ layers as $\Pi(t) = W_L(t)W_{L-1}(t)\dots W_1(t)$ and analyze the behavior of $\Pi(t)$ under the gradient flow dynamics. In particular, we derive the following result for the singular values of $\Pi(t)$.

**Theorem 1** (Informal)**.** *Assuming that the mild conditions as stated in Supplementary Materials Sec. A.3 hold. Let $\sigma_k(t)$ for $k \in [d]$ be the $k$-th largest singular value of $\Pi(t)$. Then,*

$$\dot{\sigma}_k(t) = NL\left(\sigma_k(t)\right)^{2-\frac{2}{L}} \times \sqrt{\left(\sigma_k(t)\right)^{\frac{2}{L}} + M}\left(\mathbf{u}_{L+1,k}(t)\right)^\top G(t)\mathbf{v}_k(t), \quad (4)$$

*where $\mathbf{u}_{L+1,k}(t)$ is the $k$-th left singular vector of $W_{L+1}(t)$, $\mathbf{v}_k(t)$ is the $k$-th right singular vector of $\Pi(t)$, $M$ is a constant, and $G(t)$ is defined as*

$$G(t) = \sum_{c=1}^{C} \mu_c(\mathbf{e}_c - \bar{\gamma}_c(t))\bar{X}_c^\top, \quad (5)$$

*and where $\mu_c$, $\mathbf{e}_c$, $\bar{\gamma}_c(t)$, $\bar{X}_c$ are defined after Eqn.* (3)*.*

The proof of the precise version of Theorem 1 is provided in Supplementary Materials Sec. A.

Based on Theorem 1, we are able to explain why greater data heterogeneity causes $\Pi(t)$ to be biased to become lower-rank. Note that strong data heterogeneity causes local training data of one client being highly imbalanced in terms of the number of data samples per class (recall Fig. 1(a)). This implies that $\mu_c$, which is the proportion of the class $c$ data, will be close to 0 for some classes.

Next, based on the definition of $G(t)$ in Eqn. (5), more $\mu_c$'s being close to 0 leads to $G(t)$ being biased towards a low-rank matrix. If this is so, the term $\left(\mathbf{u}_{L+1,k}(t)\right)^\top G(t)\mathbf{v}_k(t)$ in Eqn. (4) will only be significant (large in magnitude) for fewer values of $k$. This is because $\mathbf{u}_{L+1,k}(t)$ and $\mathbf{v}_k(t)$ are both singular vectors, which are orthogonal among different $k$'s. This further leads to $\dot{\sigma}_k(t)$ on the left-hand side of Eqn. (4), which is the evolving rate of $\sigma_k$, being small for most of the $k$'s throughout training. These observations imply that only relatively few singular values of $\Pi(t)$ will increase significantly after training.

Furthermore, $\Pi(t)$ being biased towards being low-rank will directly lead to dimensional collapse for the representations. To see this, we simply write the covariance matrix of the representations in terms of $\Pi(t)$ as

$$\begin{aligned}\Sigma(t) &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{z}_i(t) - \bar{\mathbf{z}}(t))(\mathbf{z}_i(t) - \bar{\mathbf{z}}(t))^\top \\ &= \Pi(t)\left(\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^\top\right)\Pi(t)^\top.\end{aligned} \quad (6)$$

From Eqn. (6), we observe that if $\Pi(t)$ evolves to being a lower-rank matrix, $\Sigma(t)$ will also tend to be lower-rank, which corresponds to the stronger dimensional collapse observed in Fig. 3.

**Algorithm 1** PyTorch-style Pseudocode for FEDDECORR.

```
# function calculating FedDecorr regularization term
# z: a batch of representation
# beta: regularization coefficient of FedDecorr
def FedDecorr(z, beta):
    # N: batch size
    # d: representation dimension
    N,d = z.shape

    # z-score normalization
    z = (z - z.mean(0)) / z.std(0)

    # estimate correlation matrix
    corr_mat = 1/N*torch.matmul(z.t(), z)

    # calculate FedDecorr loss
    loss_fed_decorr = (corr_mat.pow(2)).mean()

    return beta*loss_fed_decorr
```

## 4 MITIGATING DIMENSIONAL COLLAPSE WITH FEDDECORR

### 4.1 Method Development

Motivated by the above observations and analyses on dimensional collapse caused by data heterogeneity in federated learning, we explore how to mitigate excessive dimensional collapse.

Since dimensional collapse on the local models is the major reason for the dimensional collapse on the global model, we propose to alleviate the problem during local training. One natural way to achieve this is to add the following regularization term on the representations during training

$$L_{\text{singular}}(w, X) = \frac{1}{d}\sum_{i=1}^{d}\left(\lambda_i - \frac{1}{d}\sum_{j=1}^{d}\lambda_j\right)^2, \quad (7)$$

where $\lambda_i$ is the $i$-th singular value of the covariance matrix of the representations. Essentially, $L_{\text{singular}}$ penalizes the variance among the singular values, thus discouraging the tail singular values from collapsing to 0, mitigating dimensional collapse. However, this regularization term is not practical as it requires calculating all the singular values, which is computationally expensive.

Therefore, to derive a computationally-cheap training objective, we first apply the z-score normalization on all the representation vectors $\mathbf{z}_i$ as follows: $\hat{\mathbf{z}}_i = (\mathbf{z}_i - \bar{\mathbf{z}})/\sqrt{\text{Var}(\mathbf{z})}$. This results in the covariance matrix of $\hat{\mathbf{z}}_i$ being equal to its correlation matrix (i.e., the matrix of correlation coefficients). The following proposition suggests a more convenient cost function to regularize.

**Proposition 1.** *For a $d$-by-$d$ correlation matrix $K$ with singular values $(\lambda_1, \dots, \lambda_d)$, we have:*

$$\sum_{i=1}^{d}\left(\lambda_i - \frac{1}{d}\sum_{j=1}^{d}\lambda_j\right)^2 = \|K\|_{\text{F}}^2 - d. \quad (8)$$

The proof of Proposition 1 can be found in Supplementary Materials Sec. B. This proposition suggests that regularizing the Frobenius norm of the correlation matrix $\|K\|_{\text{F}}$ achieves the same effect as minimizing $L_{\text{singular}}$. In contrast to the singular values, $\|K\|_{\text{F}}$ can be computed efficiently.

To leverage this proposition, we propose a novel method, FEDDECORR, which regularizes the Frobenius norm of the

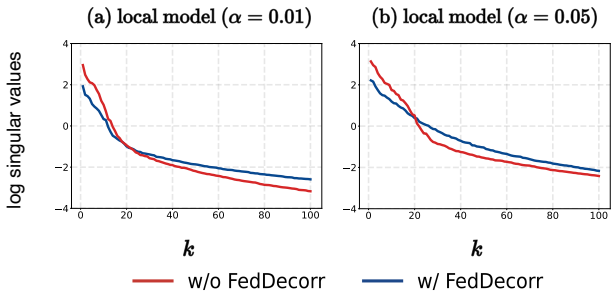



Fig. 5: FEDDECORR effectively mitigates dimensional collapse for the *local models*. For each heterogeneity parameter $\alpha \in \{0.01, 0.05\}$, we apply FEDDECORR and plot the singular values of the representation covariance matrix. The $x$-axis ($k$) is the index of singular values. With FEDDECORR, the tail singular values are prevented from dropping to 0 too rapidly.

correlation matrix of the representation vectors during local training on each client. Formally, the proposed regularization term is defined as:

$$L_{\text{FedDecorr}}(w, X) = \frac{1}{d^2}\|K\|_{\text{F}}^2, \tag{9}$$

where $w$ is the model parameters, $K$ is the correlation matrix of the representations. The overall objective of each local client is

$$\min_{w} \ell(w, X, \mathbf{y}) + \beta L_{\text{FedDecorr}}(w, X), \tag{10}$$

where $\ell$ is the cross entropy loss, and $\beta$ is the regularization coefficient of FEDDECORR. The pseudocode for calculating FEDDECORR loss is provided in Alg. 1.

### 4.2 Effectiveness of FEDDECORR

To witness the efficacy of $L_{\text{FedDecorr}}$ in mitigating dimensional collapse, we implement the method with and without $L_{\text{FedDecorr}}$ under the heterogeneous setting where $\alpha \in \{0.01, 0.05\}$ and compare the results.

Firstly, we visualize the representations of locally trained models. We plot our results in Fig. 5. As expected, applying FEDDECORR encourages the tail singular values to not collapse to 0, thus effectively mitigating dimensional collapse for the local models.

Next, we compare representations of local and global models to verify whether FEDDECORR can alleviate the dimensional collapse caused by global averaging. Our results are shown in Fig. 6. We observe that applying FEDDECORR helps to effectively reduce the $R$ defined in Eqn. 1, indicating smaller gap between curves of the singular values of the local and global models, thus mitigating the dimensional collapse from global averaging.

Finally, we visualize representations of the global model. The results are shown in Fig. 7, which illustrates that applying FEDDECORR can indeed eventually alleviate dimensional collapse for the global model.

In this section, we have shown that applying FEDDECORR can help to mitigate the two factors behind the dimensional collapse on the global model as studied in Sec. 3.2 and Sec. 3.3.

## 5 EXPERIMENTS

### 5.1 Experimental Setups

**Datasets:** In our experiment, we simulate the Federated Learning scenarios of having multiple clients for local training and one parameter server performing global aggregation. We adopt three datasets, namely CIFAR10 [54], CIFAR100 [54], and TinyImageNet [55], to evaluate under label heterogeneity settings. CIFAR10 and CIFAR100 both have $50,000$ training samples and $10,000$ test samples, and the size of each image is $32 \times 32$. TinyImageNet contains 200 classes, with $100,000$ training samples and $10,000$ testing samples, and each image is $64 \times 64$.

We use two schemes to generate local data for each client. The first method involves sampling a probability vector $\mathbf{p}_c = (p_{c,1}, p_{c,2}, \ldots, p_{c,K}) \sim \text{Dir}_K(\alpha)$ and allocating a $p_{c,k}$ proportion of instances of class $c \in [C] = \{1, 2, \ldots, C\}$ to client $k \in [K]$, where $\text{Dir}_K(\alpha)$ is the Dirichlet distribution with $K$ categories and $\alpha$ is the concentration parameter; $\alpha \downarrow 0$ means an increasing level of heterogeneity. This method follows [6], [15], [52]. The second method follows the pathological non-iid partition in [1] and assigns data of $M$ classes to each client. Under such a partition, smaller $M$ implies stronger heterogeneity. In the following sections, we refer to the first partition scheme as the Dirichlet partition and the second scheme as the pathological non-iid partition.

We use two datasets, namely Office-Caltech10 [56] and DomainNet [57], to evaluate the performance of our and competing methods under domain heterogeneity. Office-Caltech10 is a dataset acquired in different cameras or environments, containing 4 domains in total. DomainNet is a dataset with images from different styles, containing 6 domains in total. We use data of one domain as the local data of one client.

**Implementation Details:** For label heterogeneity experiments, we use a MobileNetV2 [53]. We run $100$ communication rounds for experiments on the CIFAR10/100 datasets and $50$ communication rounds on the TinyImageNet dataset. We conduct local training for $10$ epochs in each communication round using SGD optimizer with a learning rate of $0.01$, a momentum of $0.9$, and a batch size of $64$. The weight decay is set to $10^{-5}$ for CIFAR10 and $10^{-4}$ for CIFAR100 and TinyImageNet. We apply the data augmentation of [58] in CIFAR100 and TinyImageNet experiments. The $\beta$ of FEDDECORR (i.e., $\beta$ in Eqn. (10)) is tuned to be $0.1$ (See ablation study in Sec. 5.5).

In domain heterogeneity experiments, we follow the settings in [28]. Specifically, we use an AlexNet [59] with BatchNormalization [60] for all experiments. We run $300$ communication rounds for all experiments on Office-Caltech10 and DomainNet. We conduct local training for $1$ epoch in each communication round using SGD optimizer with a learning rate of $0.01$, an SGD momentum parameter of $0.9$, and a batch size of $64$. The weight decay is set to $10^{-5}$. When applying FEDDECORR to FedBN, we also directly add the FedDecorr loss in local training.

For all experiments, The regularization coefficient of FedProx [2] $\mu$ is tuned across $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and is selected to be $\mu = 10^{-3}$; the regularization coefficient of MOON [6] $\mu$ is tuned across $\{0.1, 1.0, 5.0, 10.0\}$ and is selected to be $\mu = 1.0$; the server momentum parameter of

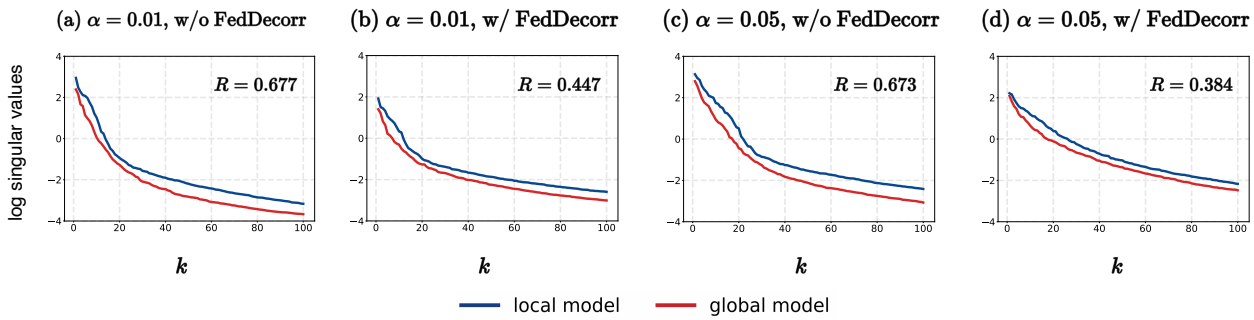



Fig. 6: FEDDECORR effectively closes the gap between the local and global models. For each heterogeneity parameter $\alpha \in \{0.01, 0.05\}$, we apply FEDDECORR and plot the singular values of the representation covariance matrix. The $x$-axis ($k$) is the index of singular values. $R$, defined in Eqn. 1, is computed for each figure and is shown on top-right corner. With FEDDECORR, the gaps between the curves of local and global models are reduced (i.e., smaller $R$).

TABLE 1: **Experiments for label heterogeneity under Dirichlet partition.** We run experiments under various degrees of heterogeneity ($\alpha \in \{0.05, 0.1, 0.5, \infty\}$) and report the test accuracy (%). All results are (re)produced by us and are averaged over 3 runs (mean $\pm$ std). Bold font highlights the highest accuracy in each column.

| Method | CIFAR10 | | | | CIFAR100 | | | | TinyImageNet | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha = 0.05$ | 0.1 | 0.5 | $\infty$ | 0.05 | 0.1 | 0.5 | $\infty$ | 0.05 | 0.1 | 0.5 | $\infty$ |
| Scaffold | 51.99±2.54 | 74.36±3.10 | 87.05±0.39 | 89.77±0.24 | 54.51±0.26 | 61.42±0.54 | 68.37±0.44 | 70.97±0.04 | 35.16±0.77 | 37.87±0.78 | 44.24±0.14 | 44.88±0.29 |
| FedNova | 63.07±1.59 | 79.98±1.56 | 90.23±0.41 | 92.39±0.18 | 60.22±0.33 | 66.43±0.26 | 71.79±0.17 | 74.47±0.13 | 35.28±0.04 | 39.73±0.07 | 47.05±0.42 | 49.57±0.09 |
| FedAvg | 64.85±2.01 | 76.28±1.22 | 89.84±0.13 | 92.39±0.26 | 59.87±0.25 | 66.46±0.16 | 71.69±0.15 | **74.54**±0.15 | 35.02±0.46 | 39.30±0.23 | 46.92±0.25 | 49.33±0.19 |
| + FEDDECORR | 73.06±0.81 | 80.60±0.91 | 89.84±0.05 | 92.19±0.10 | **61.53**±0.11 | **67.12**±0.09 | **71.91**±0.04 | 73.87±0.18 | 40.29±0.18 | 43.86±0.50 | 50.01±0.27 | 52.63±0.26 |
| FedProx | 64.11±0.84 | 76.10±0.40 | 89.57±0.04 | 92.38±0.09 | 60.02±0.46 | 66.41±0.27 | 71.78±0.19 | 74.34±0.03 | 35.20±0.30 | 39.66±0.43 | 47.16±0.07 | 49.76±0.36 |
| + FEDDECORR | 71.38±0.81 | **81.74**±0.34 | 89.96±0.26 | 92.14±0.20 | 61.33±0.19 | 67.00±0.46 | 71.64±0.10 | 74.15±0.06 | **40.63**±0.05 | 44.19±0.14 | 50.26±0.27 | 52.37±0.36 |
| FedAvgM | 71.34±0.71 | 77.51±0.58 | 88.39±0.17 | 91.35±0.15 | 59.64±0.20 | 66.36±0.14 | 71.17±0.22 | 74.20±0.16 | 34.81±0.09 | 39.72±0.11 | 47.11±0.04 | 49.67±0.25 |
| + FEDDECORR | **73.60**±0.82 | 79.21±0.15 | 88.70±0.26 | 91.33±0.13 | 61.48±0.27 | 66.60±0.11 | 71.26±0.21 | 73.86±0.25 | 39.97±0.23 | 43.95±0.26 | 50.14±0.11 | 52.05±0.37 |
| MOON | 68.79±0.69 | 78.70±0.66 | 90.08±0.10 | 92.62±0.17 | 56.79±0.17 | 65.48±0.29 | 71.81±0.14 | 74.30±0.12 | 35.23±0.26 | 40.53±0.28 | 47.25±0.66 | 50.48±0.57 |
| + FEDDECORR | 73.46±0.84 | 81.63±0.55 | **90.61**±0.05 | **92.63**±0.19 | 59.43±0.34 | 66.12±0.20 | 71.68±0.05 | 73.70±0.25 | 40.40±0.24 | **44.20**±0.22 | **50.81**±0.51 | **53.01**±0.45 |

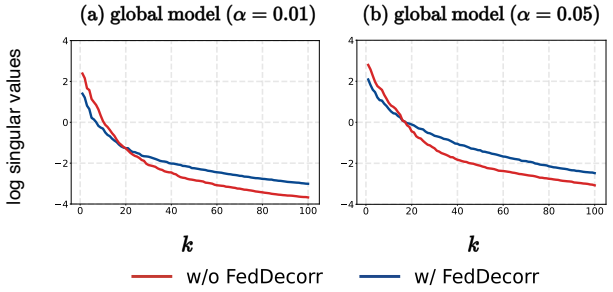



Fig. 7: FEDDECORR effectively mitigates dimensional collapse for **global models**. For each heterogeneity parameter $\alpha \in \{0.01, 0.05\}$, we apply FEDDECORR and plot the singular values of the representation covariance matrix. The $x$-axis ($k$) is the index of singular values. With FEDDECORR, the tail singular values are prevented from dropping to 0 too rapidly.

FedAvgM [13] $\rho$ is tuned across $\{0.1, 0.5, 0.9\}$ and is selected to be $\rho = 0.5$.

### 5.2 FEDDECORR Improves Baseline Methods Under Label Heterogeneity

To validate the effectiveness of our method, we apply FEDDECORR to four baselines, namely FedAvg [1], FedAvgM [13], FedProx [2], and MOON [6]. Besides, we also compare with two other baselines, namely Scaffold [3] and FedNova [4]. We first apply the Dirichlet partition (described in the second paragraph of Sec. 5.1) and split the three benchmark datasets (CIFAR10, CIFAR100, and TinyImageNet) into 10 clients with $\alpha \in \{0.05, 0.1, 0.5, \infty\}$. Since $\alpha = \infty$ corresponds to the homogeneous setting where the trained models should be free from the adverse effects of excessive dimensional collapse, we only expect FEDDECORR to perform on par with the baselines in this setting. Experimental results of Dirichlet partition are shown in Tab. 1. Next, we apply the pathological non-iid partition (also described in second paragraph of Sec. 5.1) and split the three benchmark datasets into 10 clients with $M \in \{3, 4, 5\}$ for CIFAR10, $M \in \{20, 30, 40\}$ for CIFAR100, and $M \in \{80, 90, 100\}$ for TinyImageNet. Experimental results based on the pathological non-iid partition are displayed in Tab. 2.

We observe that for all of the heterogeneous settings

TABLE 2: **Experiments for label heterogeneity under pathological non-iid partition.** We run experiments under various degrees of heterogeneity ($M \in \{3, 4, 5\}$ for CIFAR10, $M \in \{20, 30, 40\}$ for CIFAR100, $M \in \{80, 90, 100\}$ for TinyImageNet) and report the test accuracies (%). All results are (re)produced by us and are averaged over 3 runs (mean $\pm$ std). Bold font highlights the highest accuracy in each column.

| Method | CIFAR10 | | | CIFAR100 | | | TinyImageNet | | |
|---|---|---|---|---|---|---|---|---|---|
| | $M=3$ | 4 | 5 | 20 | 30 | 40 | 80 | 90 | 100 |
| Scaffold | 53.97$\pm$0.80 | 62.77$\pm$1.36 | 77.24$\pm$0.04 | 51.81$\pm$0.59 | 58.14$\pm$0.16 | 63.40$\pm$0.58 | 41.43$\pm$0.42 | 41.88$\pm$0.07 | 43.87$\pm$0.53 |
| FedNova | 59.20$\pm$0.91 | 72.54$\pm$1.54 | 85.43$\pm$0.61 | 57.95$\pm$0.22 | 63.64$\pm$0.27 | 68.79$\pm$0.29 | 41.91$\pm$0.06 | 43.96$\pm$0.17 | 45.36$\pm$0.20 |
| FedAvg | 66.64$\pm$0.63 | 74.93$\pm$1.36 | 84.69$\pm$0.70 | 58.12$\pm$0.47 | 63.43$\pm$0.14 | 67.53$\pm$1.60 | 42.14$\pm$0.43 | 43.93$\pm$0.35 | 45.61$\pm$0.06 |
| + FEDDECORR | 74.17$\pm$0.46 | 78.45$\pm$0.37 | 87.49$\pm$0.06 | 60.56$\pm$0.25 | **65.52**$\pm$0.14 | **69.57**$\pm$0.28 | 45.86$\pm$0.19 | **47.42**$\pm$0.53 | 48.52$\pm$0.25 |
| FedProx | 67.57$\pm$0.44 | 75.65$\pm$1.07 | 85.25$\pm$0.27 | 58.45$\pm$0.13 | 63.74$\pm$0.29 | 68.96$\pm$0.63 | 42.11$\pm$0.22 | 43.77$\pm$0.61 | 45.32$\pm$0.11 |
| + FEDDECORR | **75.45**$\pm$0.62 | 78.99$\pm$0.70 | 87.38$\pm$0.32 | **60.74**$\pm$0.21 | 65.39$\pm$0.43 | 69.43$\pm$0.39 | 46.20$\pm$0.58 | 47.21$\pm$0.23 | 48.55$\pm$0.04 |
| FedAvgM | 64.10$\pm$1.21 | 76.37$\pm$0.21 | 84.71$\pm$0.31 | 57.93$\pm$0.45 | 63.48$\pm$0.08 | 68.66$\pm$0.58 | 41.78$\pm$0.02 | 43.53$\pm$0.34 | 44.72$\pm$0.32 |
| + FEDDECORR | 72.69$\pm$0.32 | 77.24$\pm$0.52 | 85.85$\pm$0.04 | 60.40$\pm$0.12 | 65.25$\pm$0.21 | 69.37$\pm$0.30 | 45.45$\pm$0.20 | 46.96$\pm$0.20 | 48.41$\pm$0.14 |
| MOON | 61.04$\pm$1.47 | 76.83$\pm$0.78 | 87.63$\pm$0.32 | 54.85$\pm$0.12 | 62.70$\pm$0.21 | 68.47$\pm$0.13 | 42.73$\pm$0.11 | 44.21$\pm$0.36 | 45.82$\pm$0.17 |
| + FEDDECORR | 75.35$\pm$0.64 | **81.31**$\pm$0.31 | **88.39**$\pm$0.28 | 58.25$\pm$0.27 | 64.52$\pm$0.15 | 69.36$\pm$0.26 | **46.76**$\pm$0.29 | 47.22$\pm$0.09 | **48.80**$\pm$0.27 |

on all datasets, the highest accuracies are achieved by adding FEDDECORR on top of a certain baseline method. In particular, in the strongly heterogeneous settings where $\alpha \in \{0.05, 0.1\}$ or the smallest $M$ for each dataset, adding FEDDECORR yields significant improvements of around $\mathbf{2}\% \sim \mathbf{9}\%$ over baseline methods. On the other hand, for the less heterogeneous setting (i.e., larger $\alpha$ or $M$), the improvements from FEDDECORR are smaller. This is because the problem of dimensional collapse is less pronounced in less heterogeneous settings; this phenomenon has been discussed in Sec. 3. In addition, surprisingly, in the homogeneous setting of $\alpha = \infty$, FEDDECORR still produces around 2% of improvements on the TinyImageNet dataset. We conjecture that this is because TinyImageNet is much more complicated than the CIFAR datasets, and some other factors besides heterogeneity of label may cause undesirable dimensional collapse in the federated learning setup. Therefore, federated learning on TinyImageNet can benefit from FEDDECORR even in the homogeneous setting.

To further demonstrate the advantages of FEDDECORR, we apply it on FedAvg and plot how the test accuracy of the global model evolves with increasing communication rounds in Fig. 8. In this figure, if we set a certain value of the testing accuracy as a threshold, we see that adding FEDDECORR significantly reduces the number of communication rounds needed to achieve the given threshold. This further shows that FEDDECORR not only improves the final performance, but also greatly boosts the communication efficiency in federated learning.

### 5.3 FEDDECORR Improves Baseline Methods Under Domain Heterogeneity

Next, we evaluate the effectiveness of FEDDECORR on domain heterogeneity setting with two benchmark datasets, namely Office-Caltech10 and DomainNet. Under this setting, we assign data of one domain to be the local training data of one client. We apply FEDDECORR to five baselines, namely FedAvg [1], FedProx [2], FedAvgM [13], MOON [6], and FedBN [28]. Experiment results of domain heterogeneity are shown in Tab. 3. These results show that adding FEDDECORR yields consistent improvements over all baselines and all datasets in terms of the average accuracy by around $\mathbf{1}\% \sim \mathbf{3}\%$. Notably, FEDDECORR also demonstrates its effectiveness when being applied on a personalized method such as FedBN.

Although our analyses for FEDDECORR only focus mainly on the label heterogeneity settings, we empirically demonstrate its effectiveness can be generalized to domain heterogeneity. We conjecture this is because domain heterogeneity will also cause the trained model to suffer undesired dimensional collapse, which can be effectively alleviate by FEDDECORR. Developing a firm theoretical understanding of why FEDDECORR can counter domain heterogeneity in federated learning is left to future research.

### 5.4 Ablation Study on the Number of Clients

Next, we study whether the improvements brought by FEDDECORR are preserved as number of clients increases. We partition the TinyImageNet dataset into 10, 20, 30, 50, and 100 clients according to different $\alpha$'s, and then run FedAvg with and without FEDDECORR. For the experiments with 10, 20 and 30 clients, we run 50 communication rounds. For the experiments with 50 and 100 clients, we randomly select 20% of the total clients to participate the federated learning in each round and run 100 communication rounds. Results are shown in Tab. 4. From this table, we see that the performance improvements resulting from FEDDECORR increase from around $\mathbf{3}\% \sim \mathbf{5}\%$ to around $\mathbf{7}\% \sim \mathbf{10}\%$ with the growth in the number of clients. Therefore, interestingly, we show through experiments that the improvements brought by FEDDECORR can be even more pronounced under the more challenging settings with more clients. Moreover, our experimental results under random client participation show that the improvements from FEDDECORR are robust

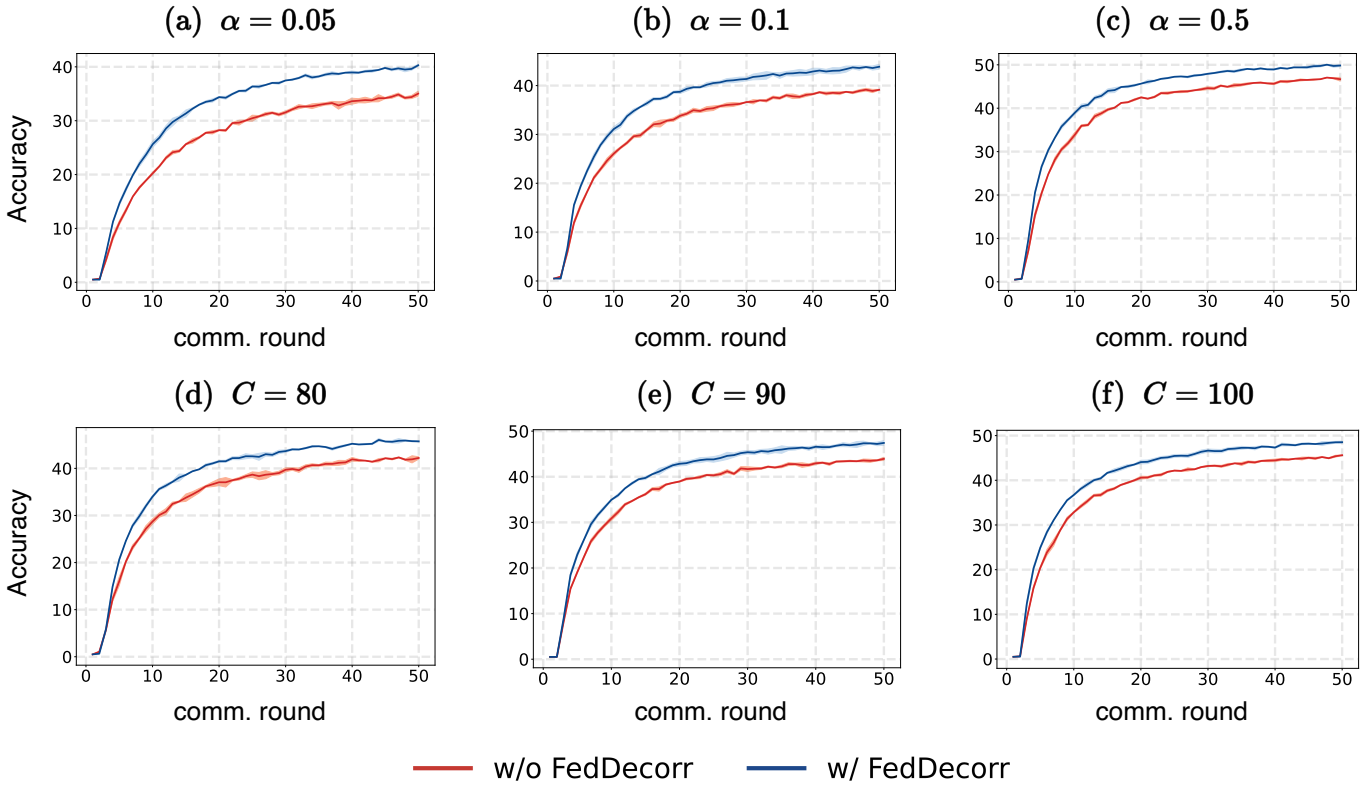



Fig. 8: **Test accuracy (%) at each communication round.** Results are based on TinyImageNet and are averaged over 3 runs. Shaded areas denote one standard deviation above and below the mean. (a-c) reports accuracy curve under Dirichlet partition with different $\alpha$'s and (d-f) reports accuracy curve under pathological non-iid partition with different $M$'s. These results show that applying FEDDECORR can consistently improve over baseline throughout the federated learning process.

to such uncertainties. These experiments demonstrate the potential of FEDDECORR to be applied to real world federated learning settings with massive numbers of clients and random client participation.

### 5.5 Ablation Study on the Regularization Coefficient $\beta$

Next, we study FEDDECORR's robustness to the $\beta$ in Eqn. (10) by varying it in the set $\{0.01, 0.05, 0.1, 0.2, 0.3\}$. We partition the CIFAR10 and TinyImageNet datasets into 10 clients with $\alpha$ equals to $0.05$ and $0.1$ to simulate the heterogeneous setting. Results are shown in Fig. 9. We observe that, in general, when $\beta$ increases, the performance of FEDDECORR first increases, then plateaus, and finally decreases slightly. These results show that FEDDECORR is relatively insensitive to the choice of $\beta$, which implies FEDDECORR is an easy-to-tune federated learning method. In addition, among all experimental setups, setting $\beta$ to be $0.1$ consistently produces (almost) the best results. Therefore, we recommend $\beta = 0.1$ when having no prior information about the dataset.

### 5.6 Ablation Study on the Number of Local Epochs

Lastly, we ablate on the number of local epochs per communication round. We set the number of local epochs $E$ to be in the set $\{1, 5, 10, 20\}$. We run experiments with and without FEDDECORR, and we use the CIFAR100 and TinyImageNet datasets with $\alpha$ being $0.05$ and $0.1$ for this ablation study. Results are shown in Tab. 5, in which one observes that with increasing $E$, FEDAVG performance first increases and then decreases. This is because when $E$ is too small, the local training cannot converge properly in each communication round. On the other hand, when $E$ is too large, the model parameters of local clients might be driven to be too far from the global optimum. Nevertheless, FEDDECORR consistently improves over the baselines across different choices of local epochs $E$.

### 5.7 Computational Efficiency

We demonstrate FEDDECORR's advantage vis-à-vis some of its competitors in terms of their computational efficiencies. We compare FEDDECORR with some other methods that also apply additional regularization terms during local training such as FedProx and MOON. We partition CIFAR10, CIFAR100, and TinyImageNet into 10 clients with $\alpha = 0.5$ and report the total computation times required for one round of training for FedAvg, FedProx, MOON, and FEDDECORR. Specifically, for FedAvg, only naïve SGD is applied during local training. Results are shown in Tab. 6. All results are produced with a NVIDIA Tesla V100 GPU. We see that FEDDECORR incurs a negligible computation overhead on top of the vanilla FedAvg, while FedProx and MOON require about $0.5 \sim 1$ times additional computation cost. The advantage of FEDDECORR in terms of efficiency is mainly because it involves calculating only the Frobenius norm of a matrix, which is extremely cheap. Indeed, this regularization operates on the output representation vectors

TABLE 3: **Experiments for Domain Heterogeneity.** We assign data from one domain as local data of one client. All results are (re)produced by us and are averaged over 3 runs (mean±std). For Office-Caltech10, we report the average accuracy over all domains and accuracy on each domain (i.e., Amazon, Caltech, Dslr, Webcam). For DomainNet, we report the average accuracy over all domains and accuracy on each domain (i.e., Clipart, Infograph, Painting, Quickdraw, Real, Sketch). Bold font highlights the highest accuracy in each column separately for non-personalized methods personalized methods (FedBN).

| Method | Office-Caltech10 | | | | | DomainNet | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg | A | C | D | W | Avg | C | I | P | Q | R | S |
| FedAvg | 64.51±0.93 | 50.52±0.74 | 45.93±0.76 | 72.91±3.90 | 88.70±0.80 | 42.50±0.71 | 50.19±2.44 | 26.18±1.20 | 39.74±2.60 | 53.57±2.04 | 48.20±0.80 | 37.12±0.55 |
| + FEDDECORR | 66.19±0.91 | 55.56±0.65 | **47.56**±1.67 | 72.92±1.47 | 88.70±1.60 | 43.69±0.71 | 51.33±0.41 | **27.50**±1.57 | 38.18±1.27 | 56.00±2.48 | 48.56±1.95 | **40.56**±1.37 |
| FedProx | 64.61±1.05 | 51.73±0.25 | 47.11±0.63 | 69.79±3.83 | 89.83±1.38 | 42.59±0.42 | 50.00±0.62 | 26.69±0.38 | 38.72±1.64 | 55.13±3.95 | 48.07±1.98 | 36.95±1.53 |
| + FEDDECORR | 65.71±2.57 | 54.17±1.70 | 45.33±2.38 | 72.92±5.31 | 90.40±1.60 | 43.97±0.74 | **52.03**±1.65 | 27.14±0.64 | 39.42±0.23 | 57.20±2.78 | 48.70±0.61 | 39.35±1.35 |
| FedAvgM | 64.54±0.91 | 50.69±0.88 | 45.48±0.42 | 75.00±2.55 | 87.01±2.11 | 41.82±0.65 | 50.19±0.16 | 25.06±0.80 | 38.50±0.93 | 55.17±2.28 | 45.71±0.45 | 36.28±1.56 |
| + FEDDECORR | 66.08±1.14 | 54.51±1.77 | 46.67±1.58 | 75.01±5.10 | 88.13±1.38 | **44.07**±1.08 | 51.90±0.27 | 25.77±0.69 | **40.33**±0.77 | **58.17**±2.32 | **49.22**±1.94 | 39.05±1.19 |
| MOON | 66.43±0.84 | 54.34±1.36 | 46.67±0.96 | 76.04±1.48 | 88.70±1.60 | 40.51±0.47 | 47.78±1.48 | 25.06±0.90 | 36.14±2.97 | 51.87±2.13 | 46.81±0.81 | 35.38±0.78 |
| + FEDDECORR | **67.69**±0.68 | **55.90**±0.98 | 45.78±1.26 | **78.13**±0.48 | **90.96**±0.80 | 41.16±0.50 | 49.11±0.91 | 25.72±0.98 | 37.10±0.28 | 51.77±2.78 | 46.53±0.85 | 36.71±1.87 |
| FedBN | 72.73±0.50 | **63.54**±0.73 | 45.03±0.21 | 88.54±1.48 | **93.79**±2.88 | 48.16±0.63 | 52.22±0.85 | **28.21**±0.38 | **42.54**±0.20 | 70.30±0.45 | **56.67**±1.31 | 38.99±2.17 |
| + FEDDECORR | **74.00**±0.51 | 62.67±0.88 | **48.45**±0.96 | **91.67**±1.45 | 93.22±1.38 | **48.36**±0.21 | **52.85**±0.94 | 27.75±0.73 | 40.82±0.88 | **71.33**±0.66 | 54.83±0.94 | **42.54**±1.12 |

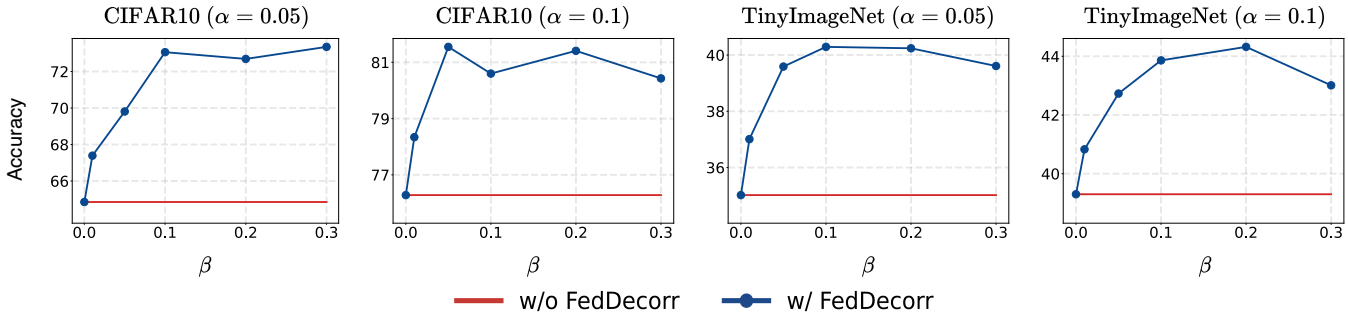



Fig. 9: **Ablation study on $\beta$.** We apply FEDDECORR with different choices of $\beta$ on FedAvg. These results show that improvements brought by FEDDECORR is robust to the choice of $\beta$ and a good rule-of-thumb is to choose $\beta = 0.1$.

TABLE 4: **Ablation study on the number of clients.** Based on CIFAR100 and TinyImageNet, we run experiments with different number of clients and different amount of data heterogeneity.

| # clients | Method | CIFAR100 | | TinyImageNet | |
|---|---|---|---|---|---|
| | | $\alpha = 0.05$ | 0.1 | 0.05 | 0.1 |
| 10 | FedAvg | 59.87 | 66.46 | 35.02 | 39.30 |
| | + FEDDECORR | 61.53 | 67.12 | 40.29 | 43.86 |
| 20 | FedAvg | 59.56 | 62.78 | 31.21 | 35.30 |
| | + FEDDECORR | 61.31 | 63.30 | 39.41 | 41.27 |
| 30 | FedAvg | 56.00 | 61.77 | 26.20 | 30.88 |
| | + FEDDECORR | 57.47 | 62.54 | 36.50 | 39.02 |
| 50 | FedAvg | 44.66 | 53.28 | 25.70 | 28.88 |
| | + FEDDECORR | 48.06 | 54.48 | 34.50 | 36.67 |
| 100 | FedAvg | 41.40 | 50.45 | 21.53 | 24.69 |
| | + FEDDECORR | 46.78 | 51.90 | 30.55 | 33.85 |

TABLE 5: **Ablation study on local epochs.** Experiments with different number of local epochs.

| # Local Epochs | Method | CIFAR100 | | TinyImageNet | |
|---|---|---|---|---|---|
| | | $\alpha = 0.05$ | 0.1 | 0.05 | 0.1 |
| 1 | FedAvg | 50.67 | 55.98 | 32.31 | 34.88 |
| | + FEDDECORR | 53.18 | 57.02 | 36.49 | 38.99 |
| 5 | FedAvg | 59.57 | 65.02 | 36.02 | 40.75 |
| | + FEDDECORR | 61.42 | 65.98 | 41.68 | 44.77 |
| 10 | FedAvg | 59.87 | 66.46 | 35.02 | 39.30 |
| | + FEDDECORR | 61.53 | 67.12 | 40.29 | 43.86 |
| 20 | FedAvg | 58.50 | 66.37 | 31.23 | 37.23 |
| | + FEDDECORR | 60.65 | 66.86 | 35.44 | 42.04 |

of the model, neither requiring the computing of parameter-wise regularization like FedProx nor extra forward passes like MOON.

TABLE 6: **Comparison of computation times.** We report the total computation times (in minutes) for one round of training on the three datasets for FedAvg, FedProx, MOON, and FEDDECORR. Here, FEDDECORR stands for applying FEDDECORR to FedAvg.

| | CIFAR10 | CIFAR100 | TinyImageNet |
|---|---|---|---|
| FedAvg | 6.7 | 6.9 | 25.4 |
| FedProx | 12.1 | 12.3 | 33.2 |
| MOON | 12.2 | 12.7 | 38.1 |
| FEDDECORR | 6.9 | 7.1 | 25.7 |

### 5.8 Comparison with Other Decorrelation Methods

Some decorrelation regularizations such as DeCov [49] and Structured-DeCov [51] were proposed to improve the generalization capabilities in standard classification tasks. Both these methods operate directly on the covariance matrix of the representations instead of the correlation matrix like our proposed method—FEDDECORR. To compare our FEDDECORR with the existing decorrelation methods, we follow the same procedure as in FEDDECORR and apply DeCov and Structured-DeCov during local training. Our experiments are based on TinyImageNet and FedAvg. TinyImageNet is partitioned into 10 clients according to various $\alpha$'s. Results are shown in Tab. 7. Surprisingly, we see that unlike our FEDDECORR which steadily improves the baseline, adding DeCov or Structured-DeCov both degrade the performance in federated learning. We conjecture that this is because directly regularizing the covariance matrix may be highly unstable, leading to undesired modification on the representations. This experiment shows that our design of regularization of the *correlation matrix* instead of the *covariance matrix* is of paramount importance to ensure stability.

TABLE 7: **Comparison with other decorrelation methods.** Based on FedAvg and the TinyImageNet dataset, we use different decorrelation regularizers in local training. Bold font highlights the highest accuracy for each $\alpha$.

| | FedAvg | DeCov | St.-Decov | FEDDECORR |
|---|---|---|---|---|
| $\alpha = 0.05$ | 35.02 | 32.88 | 32.04 | **40.29** |
| $\alpha = 0.1$ | 39.30 | 37.29 | 37.74 | **43.86** |
| $\alpha = 0.5$ | 46.92 | 46.29 | 45.85 | **50.01** |

### 5.9 Experiments on Other Model Architectures

In this section, we demonstrate the effectiveness of FEDDECORR across different model architectures. Here, in additino to the MobileNetV2 used in previous sections, we also experiment on ResNet18 and ResNet32 [61]. Note that ResNet18 is the wider version of ResNet whose representation dimension is 512 and ResNet32 is the narrower version whose representation dimension is only 64. The heterogeneity parameter $\alpha$ is set to be $\{0.05, 0.1\}$ and we use the CIFAR10 dataset. Our results are shown in Tab. 8. As can be seen, FEDDECORR yields consistent improvements across different neural network architectures. One interesting phenomenon is that the improvements brought about by FEDDECORR are much larger on wider networks (e.g., MobileNetV2, ResNet18) than on narrower ones (e.g. ResNet32). We conjecture that this is because the dimension of the ambient spaces of wider networks are clearly higher than that of shallower networks. Therefore, relatively speaking, the dimensional collapse caused by data heterogeneity will be more severe for wider networks.

TABLE 8: **Effectiveness of FEDDECORR on other model architectures.**

| | MobileNetV2 | ResNet18 | ResNet32 |
|---|---|---|---|
| FedAvg ($\alpha = 0.05$) | 64.85 | 71.51 | 65.76 |
| + FEDDECORR ($\alpha = 0.05$) | 73.06 | 76.54 | 67.21 |
| FedAvg ($\alpha = 0.1$) | 76.28 | 82.32 | 73.22 |
| + FEDDECORR ($\alpha = 0.1$) | 80.60 | 83.59 | 74.75 |

## 6 CONCLUSIONS

In this work, we study representations of trained models under federated learning in which the data held by clients are heterogeneous. Through extensive empirical observations and theoretical analyses, we show that stronger data heterogeneity results in more severe dimensional collapse for both global and local representations. Motivated by this, we propose FEDDECORR, a novel method to mitigate dimensional collapse, thus improving federated learning under the heterogeneous data setting. Extensive experiments on benchmark datasets show that FEDDECORR yields consistent improvements over existing baseline methods.

## REFERENCES

[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] 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.

[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 objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.

[5] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," *arXiv preprint arXiv:2102.02079*, 2021.

[6] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.

[7] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[8] M. Al-Shedivat, J. Gillenwater, E. Xing, and A. Rostamizadeh, "Federated learning via posterior averaging: A new perspective and practical algorithms," *arXiv preprint arXiv:2010.05273*, 2020.

[9] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 10 165–10 173.

[10] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 250–18 280.

[11] D. Caldarola, B. Caputo, and M. Ciccone, "Improving generalization in federated learning by seeking flat minima," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*. Springer, 2022, pp. 654–672.

[12] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8397–8406.

[13] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[14] 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, 2021.

[15] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[16] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2351–2363, 2020.

[17] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[18] D. Jhunjhunwala, S. Wang, and G. Joshi, "Fedexp: Speeding up federated averaging via extrapolation," *arXiv preprint arXiv:2301.09604*, 2023.

[19] X.-C. Li and D.-C. Zhan, "Fedrs: Federated learning with restricted softmax for label distribution non-iid data," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 995–1005.

[20] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 311–26 329.

[21] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[22] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[23] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3557–3568, 2020.

[24] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 394–21 405, 2020.

[25] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik, "Lower bounds and optimal algorithms for personalized federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2304–2315, 2020.

[26] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 7865–7873.

[27] M. Zhang, K. Sapra, S. Fidler, S. Yeung, and J. M. Alvarez, "Personalized federated learning with first order model optimization," *arXiv preprint arXiv:2012.08565*, 2020.

[28] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[29] Y. Liu, Y. Kang, T. Zou, Y. Pu, Y. He, X. Ye, Y. Ouyang, Y.-Q. Zhang, and Q. Yang, "Vertical federated learning," *arXiv preprint arXiv:2211.12814*, 2022.

[30] Y. Liu, Y. Kang, X. Zhang, L. Li, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient collaborative learning framework for distributed features," *arXiv preprint arXiv:1912.11187*, 2019.

[31] F. Fu, X. Miao, J. Jiang, H. Xue, and B. Cui, "Towards communication-efficient vertical federated learning training via cache-enabled local updates," *arXiv preprint arXiv:2207.14628*, 2022.

[32] T. J. Castiglia, A. Das, S. Wang, and S. Patterson, "Compressed-vfl: Communication-efficient learning with vertically partitioned data," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2738–2766.

[33] S. P. Singh and M. Jaggi, "Model fusion via optimal transport," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 045–22 055, 2020.

[34] C. Liu, C. Lou, R. Wang, A. Y. Xi, L. Shen, and J. Yan, "Deep neural network fusion via graph matching with applications to model ensemble and federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 13 857–13 869.

[35] T. Uriot and D. Izzo, "Safe crossover of neural networks through neuron alignment," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 435–443.

[36] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 965–23 998.

[37] S. K. Ainsworth, J. Hayase, and S. Srinivasa, "Git re-basin: Merging models modulo permutation symmetries," *arXiv preprint arXiv:2209.04836*, 2022.

[38] S. Ashmore and M. Gashler, "A method for finding similarity between multi-layer perceptrons by forward bipartite alignment," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[39] K. Roth, T. Milbich, S. Sinha, P. Gupta, B. Ommer, and J. P. Cohen, "Revisiting training strategies and generalization performance in deep metric learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8242–8252.

[40] L. Jing, P. Vincent, Y. LeCun, and Y. Tian, "Understanding dimensional collapse in contrastive self-supervised learning," *arXiv preprint arXiv:2110.09348*, 2021.

[41] T. Hua, W. Wang, Z. Xue, S. Ren, Y. Wang, and H. Zhao, "On feature decorrelation in self-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9598–9608.

[42] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, "Whitening for self-supervised representation learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3015–3024.

[43] Y. Shi, K. Zhou, J. Liang, Z. Jiang, J. Feng, P. H. Torr, S. Bai, and V. Y. Tan, "Mimicking the oracle: an initial phase decorrelation approach for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 722–16 731.

[44] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization," in *International Conference on Machine Learning*. PMLR, 2018, pp. 244–253.

[45] S. Arora, N. Cohen, W. Hu, and Y. Luo, "Implicit regularization in deep matrix factorization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[46] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 268–10 278.

[47] A. Bardes, J. Ponce, and Y. LeCun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," *arXiv preprint arXiv:2105.04906*, 2021.

[48] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 310–12 320.

[49] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," *arXiv preprint arXiv:1511.06068*, 2015.

[50] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 791–800.

[51] W. Xiong, B. Du, L. Zhang, R. Hu, and D. Tao, "Regularizing deep convolutional neural networks with a structured decorrelation constraint," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 519–528.

[52] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7252–7261.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

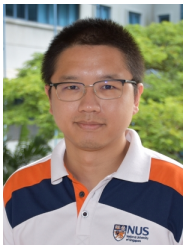
[54] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[55] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
[56] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE, 2012, pp. 2066–2073.
[57] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1406–1415.
[58] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.
[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[60] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.
[61] 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*, 2016, pp. 770–778.

**Yujun Shi** is currently a PhD student in Department of Electrical and Computer Engineering, National University of Singapore, advised by Assoc. Prof. Vincent Y. F. Tan. He received the B.Eng. in Computer Science from Nankai University in 2019. His research interests focus on tackling distribution shift scenarios in deep learning such as Continual Learning, Federated Learning, etc.

**Jian Liang** (Member, IEEE) received the B.E. degree in Electronic Information and Technology from Xi'an Jiaotong University and Ph.D. degree in Pattern Recognition and Intelligent Systems from NLPR, CASIA in July 2013, and January 2019, respectively. He was a research fellow at the National University of Singapore from June 2019 to April 2021. Now he joins NLPR as an associate professor. His research interests focus on transfer learning, pattern recognition, and computer vision.

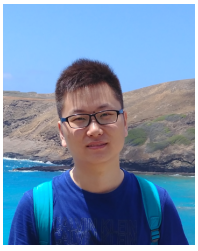
**Wenqing Zhang** received his M.S. degree in Information and Communication Engineering from Huazhong University of Science and Technology (HUST), China. He received his B.S. degree in Telecommunications Engineering from Xidian University, China. His research interest mainly focuses on computer vision, including multi-modality scene understanding, text recognition and detection.

**Chuhui Xue** is currently a computer vision scientist in ByteDance, Singapore. She received her Ph.D and B.Eng. in School of Computer Science and Engineering, Nanyang Technological University, Singapore. She works on scene text detection, scene text recognition and document analysis.



**Vincent Y. F. Tan** (S'07-M'11-SM'15) was born in Singapore in 1981. He is currently an Associate Professor in the Department of Mathematics and the Department of Electrical and Computer Engineering at the National University of Singapore (NUS). He received the B.A. and M.Eng. degrees in Electrical and Information Sciences from Cambridge University in 2005 and the Ph.D. degree in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology (MIT) in 2011. His research interests include information theory, machine learning, and statistical signal processing.

Dr. Tan received the MIT EECS Jin-Au Kong outstanding doctoral thesis prize in 2011, the NUS Young Investigator Award in 2014, the Singapore National Research Foundation (NRF) Fellowship (Class of 2018) and the NUS Young Researcher Award in 2019. He was also an IEEE Information Theory Society Distinguished Lecturer for 2018/9. He is currently serving as a Senior Area Editor of the *IEEE Transactions on Signal Processing* and an Associate Editor of Machine Learning for the *IEEE Transactions on Information Theory*. He is a member of the IEEE Information Theory Society Board of Governors.



**Dr. Song Bai** is a Computer Vision Lead in ByteDance/TikTok Singapore, and also holds an appointment as an Adjunct Assistant Professor at the Department of Electrical and Computer Engineering, National University of Singapore. Before that, he was a research fellow at the University of Oxford working with Prof. Philip H.S. Torr. He serves as an Associate Editor of Pattern Recognition and a Guest Editor of IEEE Transactions on Pattern Analysis and Machine Intelligence. He has served as Area Chair/SPC of CVPR 2023, NeurIPS 2023, and AAAI 2022. His research interests include computer vision and machine learning, especially video understanding.