FEDCUAU: CLUSTERED FEDERATED LEARNING USING WEIGHT DIVERGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

The majority of federated learning (FL) approaches aim to learn either a high-performing global model or multiple personalized models. Although there has been significant progress in each research direction, the optimization of one often comes at the expense of the other. In this work, we approach this problem by investigating how different clusters of clients with varying degrees of data heterogeneity may impact the single global model. From this analysis, we discover a surprising insight: despite a significant distribution mismatch between clusters, the knowledge shared from low data heterogeneous clusters to high data heterogeneous clusters can significantly boost the latter's personalized accuracy but not vice versa. By building on this observation, we propose a cluster-based approach named FedCUAU, in which clients are clustered based on their degree of data heterogeneity, and knowledge between each cluster is selectively transferred. We also offer provable assurance to show that FedCUAU can be used to accurately and efficiently cluster clients in a one-shot manner. Experimental results on standard FL benchmarks show that FedCUAU can be plugged into existing FL algorithms to achieve considerable improvement both the initial and personalized performance. Empirical results shows that FedCUAU improves FedAvg initial global accuracy by 1.53% and 1.82% for CIFAR10 and FEMNIST respectively, and personalized accuracy by 0.29% and 3.81%.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017) has become an indispensable tool to enable privacy-preserving collaborative learning in order to improve a single global model or to deliver better personalized models tailored to the end-user's local data and context (Arivazhagan et al., 2019; Hilmkil et al., 2021; Cheng et al., 2021). One of the main challenges of FL is to build high-performing models given the label distribution skew where the marginal distributions of labels vary drastically across clients. Specifically, as the degree of data heterogeneity among clients (non-IIDness) increases or decreases, the initial performance of a single global model significantly degrades or improves respectively (Yurochkin et al., 2019; Qiu et al., 2022). On the contrary, a greater extent of non-IIDness leads to increased in personalized performance as the local task is easier, with more samples per class, for individual clients to optimize on (Jiang et al., 2019). Although both metrics are important to cater to different scenarios, the majority of existing works aim to improve performance on either.

Due to the opposing effects of data heterogeneity, optimizing for both initial and personalized performance using a single global model often comes at a trade-off. For instance, Jiang et al. (2019) showed that improving the initial accuracy of the global model would hurt the model's capacity to personalize. Hence, existing solutions (Oh et al., 2021; Mansour et al., 2020; Matsuda et al., 2022a) mitigated the problem through model decoupling, in which the global model is split into shared and personalized layers, and/or using multiple models, each of which is catered to each client or each cluster of clients. Notably, Oh et al. (2021) showed that model decoupling can lead to better representation in the shared layers, resulting in a higher initial accuracy, and better personalized accuracy after fine-tuning the personalized layers. However, unlike model decoupling, the impact of the collaborative learning among multiple models on both metrics is still not well understood.

In this work, we first study the impact of data heterogeneity on a single global model in a clustered FL setting by experimenting with clusters of clients of varying degrees of data heterogeneity. Through this ablation study, we observe that clusters with a higher degree of non-IIDness can benefit from the collaboration with clusters with a lower degree of non-IIDness despite the significant distribution mismatch, whereas the reverse is often detrimental (Section 4). Motivated by this result, we then propose leveraging weight divergence between each client's update (*CU*) and their aggregated update (*AU*) in a single FL training round as an effective metric, named *CUAU*, to cluster clients based on their degree of data heterogeneity without the collection of any additional information (Section 5). Not only do we differ from other clustering approach by technique design, but more importantly, the solution offers provable assurance of behaviour (Section 5). Lastly, we present a simple clustered and training FL algorithm, named FedCUAU, which clusters clients based on our proposed *CUAU* and holds explicit rules in governing the knowledge shared among clusters (Section 6).

Unlike previous clustered FL works, in which cluster is based on similar data distributions, FedCUAU clusters based on the degree of non-IIDness. This offers a couple of key advantages: *1)* enables accurate cluster identity estimations to be performed in an efficient one-shot manner and *2)* allows a greater control over the impact of training with non-IID data on both the initial and personalized performance. Empirical results on federated image classification show that FedCUAU can be combined with existing FL methods for considerable gains in both the initial accuracy of a single global model and the personalized accuracy of local models. Experimental results shows that FedCUAU improves FedAvg initial global accuracy by 1.53% and 1.82% for CIFAR10 and FEMNIST respectively, and personalized accuracy by 0.29% and 3.81%. More detailed results can be found in Section 6.3 for other FL methods.

2 Related Work

Global Model FL. Since FedAvg (McMahan et al., 2017) was proposed as a practical alternative to FedSGD (Shokri & Shmatikov, 2015) to FL, there has been a surge of works that aim to tackle the significant performance degradation when applying FedAvg in non-IID settings. Zhao et al. (2018) showed that this performance drop could be attributed to the difference of weights, starting from the same weight initialization, between a model trained using FedAvg and a model trained centrally using SGD. They proved that this weight divergence is bounded by the earth mover's distance (EMD) which increases as the degree of non-IIDness increases, and proposed sharing a small percentage of IID data for local training to reduce this divergence, hence improve performance. Apart from data sharing, FedProx (Li et al., 2020) used a regularization term to minimize the Euclidean distance between the local model and the global model. Besides Euclidean distance, MOON (Li et al., 2021a) reduced the weight divergence by maximizing the agreement of the local model's and global model's representations using contrastive learning. SCAFFOLD (Karimireddy et al., 2020) proposed using control variates to shift the local model update towards the estimated true optimum trained using centralized SGD. CCVR (Luo et al., 2021) decoupled the model and showed that fine-tuning the classifier using IID features can improve initial accuracy regardless of the degree of data heterogeneity. To preserve privacy, the authors used features drawn from a Gaussian Mixture Model, which is estimated using the clients' local data statistics.

Personalized FL. The goal of personalized FL is to maximize the local model's performance on each client. To this end, fine-tuning from a global model locally on each client can achieve competitive results (Matsuda et al., 2022a;b; Chen et al., 2022; Jiang et al., 2019). Nonetheless, dedicated works have been proposed to improve personalization better. FedMe (Matsuda et al., 2022a) and FML (Shen et al., 2020) used deep mutual learning (Zhang et al., 2018) to transfer knowledge between the local and global model. Similar to FedProx, Ditto (Li et al., 2021b) utilized a L_2 term between the local models and global model. Apart from L_2, the use of Moreau envelopes has also been shown to be effective in pFedMe (T Dinh et al., 2020). Meta-learning algorithms such as MAML (Finn et al., 2017) have also been adopted in Per-FedAvg (Fallah et al., 2020) and FedMeta (Chen et al., 2018) for local optimization. Besides algorithmic changes, many works proposed decoupling the model layers into shared and personalized layers. FedPer (Arivazhagan et al., 2019), FedRep (Collins et al., 2021), and FedBABU (Oh et al., 2021) shared the earlier layers in order to learn better representations and kept the deeper layers private to each client as the deeper layers have been shown to be easily biased to the local distribution (Luo et al., 2021; Zhuang et al.,

2021). Conversely, LG-FedAvg (Liang et al., 2020) demonstrated substantial efficiency gains by only sharing the deeper layers and thus learning the representations locally.

Clustered FL. Clustered FL typically entails grouping clients into clusters, where each cluster learns its own model in a federated manner. In HypCluster (Mansour et al., 2020), the client is assigned the cluster model with the lowest loss when evaluated over its own local data. Therefore, each client has to evaluate models in all clusters every round. To reduce communication costs, IFCA (Ghosh et al., 2020) proposed clustering only the deeper layers of the model. Besides estimating cluster identities on the client-end, numerous works proposed to cluster on the server given the clients' trained local model or gradients either in a one-shot or iterative manner. Ghosh et al. (2019) and FedMe (Matsuda et al., 2022a) applied K-means to the weights of the clients' models or to the outputs of these models using unlabeled data respectively. CFL (Sattler et al., 2020) utilized the cosine similarity while FL+HC (Briggs et al., 2020) adopted various distance metrics between the clients' gradients for clustering. Ultimately, these works aim to group clients with similar data distributions within a cluster. To allow knowledge transfers among clusters, FedFOMO (Zhang et al., 2021) learns a weighted sum of other clients' local models for each local model, and FedEM Marfoq et al. (2021) learns a weighted sum of a set of multiple models representing independent distributions.

3 PRELIMINARIES

3.1 PROBLEM SETUP

In this section, we formally define the problem of centralized federated image classification. We consider a L class classification problem defined over a compact space $\mathcal{X}$ and a label space $\mathcal{Y} = [L]$, where $[L] = \{1, ..., L\}$. The data point $\{\mathbf{x}, y\}$ distributes over $\mathcal{X} \times \mathcal{Y}$ following the distribution p. A function $f : \mathcal{X} \to \mathcal{S}$ maps $\mathbf{x}$ to the probability simplex $\mathcal{S}$, $\mathcal{S} = \{\mathbf{z} | \sum_{i=1}^{L} z_i = 1, z_i \geq 0, \forall i \in [L]\}$ with f_i denoting the probability for the ith class. f is parameterized over the hypothesis class w, which is the weights of the neural network. $\mathcal{L}(\mathbf{w})$ is the loss function, and we assume the widely used cross-entropy loss:

$$\mathcal{L}(\mathbf{w}) = \mathbb{E}[\sum_{i=1}^{L} \mathbb{I}_{y=i} \log f_i(\mathbf{x}, \mathbf{w})] = \sum_{i=1}^{L} p(y = i)\mathbb{E}[\log f_i(\mathbf{x}, \mathbf{w})]. \tag{1}$$

FL methods are designed to handle multiple devices[1] collecting data and a central server coordinating the global learning objective across the network. Assume there are total N devices in the clients pool, and K clients are randomly selected for local training in each round, with E steps of local update. In particular, the objective of FL is to minimize:

$$\min_{w} f(w) = \sum_{k=1}^{N} q_k F_k(w) = \mathbb{E}[F_k(w)] \tag{2}$$

where $q_k \geq 0$ and $\sum_k q_k = 1$. The local objectives measure the local empirical risk over possibly differing data distribution $\mathcal{D}_k$, i.e., $F_k(w) = \mathbb{E}_{x_k \sim \mathcal{D}_k}[f_k(w, x_k)]$, with n_k samples available at each device k. Hence, q_k is set as $q_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ is the total data samples.

Let w_0 be a set of randomly initialized weights. We then further define $w_t = \sum_{k=1}^{K} w_{t,k}$ to be the aggregated weight at round t where $w_{t,k}$ is the weight of client k. Hence, the client update (*CU*) for client k at round t is $CU_{t,k} = w_{t,k} - w_{t-1}$ and the aggregated update (*AU*) for all K participating clients is $AU_t = w_t - w_{t-1}$. After training for T rounds, the objective of personalization is to minimize $f(w_T)$ using local dataset $x_k \sim \mathcal{D}_k$. In our work, we focus on optimizing both the initial global performance and the personalized performance for each client.

3.2 EXPERIMENTAL SETUP

Datasets & Data Partitioning. Experiments are conducted on two image classification tasks of different complexity: CIFAR10 (Krizhevsky et al., 2009) and FEMNIST (Caldas et al., 2018). FEMNIST consists of a total of 3597 clients of varying number of samples and class labels as defined

[1] we use the terms devices and clients interchangeably.

Table 1: Initial and personalized accuracy of a single global model trained by different predefined clusters using FedAvg on CIFAR10. We report the mean and standard deviation of the initial accuracy and the personalized accuracy for each cluster, along with the average personalized accuracy across clusters, across 3 separate runs.

Train Clusters	Initial Acc	C1 Per. Acc	C2 Per. Acc	C3 Per. Acc	C4 Per. Acc	Per. Acc
Incrementally adding Clusters with a Greater Degree of Non-IIDness						
C1	82.76±0.39	82.29±0.26	83.35±0.81	81.56±0.74	81.84±0.39	82.26±0.18
C1+C2	**83.97±0.25**	**83.77±0.20**	85.04±0.14	83.95±0.67	87.17±0.54	84.98±0.28
C1+C2+C3	83.85±0.21	83.57±0.23	85.15±0.13	84.77±0.36	89.97±0.12	85.87±0.13
C1+C2+C3+C4	82.68±0.07	82.80±0.62	**85.91±0.16**	**85.55±0.31**	**92.95±0.06**	**86.80±0.13**
Incrementally adding Clusters with a Lower Degree of Non-IIDness						
C4	63.22±0.62	68.69±0.58	74.76±0.25	76.77±0.40	91.31±0.32	77.88±0.37
C3+C4	77.27±0.61	78.03±0.26	82.60±0.66	83.47±1.18	92.61±0.40	84.18±0.54
C2+C3+C4	80.76±0.33	81.04±0.16	84.64±0.36	84.83±0.68	92.72±0.16	85.81±0.25
C1+C2+C3+C4	**82.68±0.07**	**82.80±0.62**	**85.91±0.16**	**85.55±0.31**	**92.95±0.06**	**86.80±0.13**

in Caldas et al. (2018). For CIFAR10, we set of the number of clients to 100 and follow the latent Dirichlet allocation (LDA) partition method (Hsu et al., 2019; Yurochkin et al., 2019; Qiu et al., 2022), allocating the same number of train and test samples to each client. Specifically, we draw local class labels $y \sim Dir(\alpha)$ for each client to form local distribution p, hence, the degree of data heterogeneity is parameterized by α. As $\alpha \to \infty$, local class labels become more uniform (IID), and as $\alpha \to 0$, these labels become less uniform (non-IID).

Model Architecture & Training Details. Following previous work (Horvath et al., 2021), a ResNet-18 (He et al., 2016) architecture is used for CIFAR-10. For FEMNIST, we employed the CNN first proposed in (Caldas et al., 2018). For both datasets, the models are trained with SGD and the number of local client epochs is set to 1. For CIFAR-10, the starting learning rate, η_1, for CIFAR-10 and FEMNIST is set to 0.1 and 0.01 and the total number of FL rounds, T is set to 500 and 200 respectively. For CIFAR-10, we dropped η by 0.1 at round 250 and 375, and for FEMNIST, an exponential LR decay is set per round: $\eta_t = \eta_1 \exp(\frac{t}{T} \log(\eta_1/\eta_T))$ where the learning rate of the last round is $\eta_T = 0.006$. The number of participating clients per round, K, is set to 10 and 35 for CIFAR-10 and FEMNIST respectively. Lastly, we fine-tune each client using 5 epochs with LR η_T starting from the trained FL model, w_T, to obtain the personalized accuracy on all experiments.

4 Impact of Data Heterogeneity on Clusters' Performance

In this section, we study the impact of different clusters of varying data heterogeneity on a single global model using FedAvg McMahan et al. (2017). In many previous FL works Hsu et al. (2019); Yurochkin et al. (2019); Qiu et al. (2022), the entire dataset is partitioned using a single parameter α to indicate the degree of non-IIDness as described in Section 3.2. Instead, we divide the CIFAR10 dataset, along with the number of clients, uniformly into four clusters and use a different $alpha$ for each cluster as shown in Table 2; *e.g.* clients in C1 have a label distribution close to uniform and clients in C4 have a highly skewed label distribution. This results in a wider range of marginal label distributions at a global level with varying degrees of non-IIDness in each cluster, a scenario that is common in real-world scenarios.

Table 2: Degree of non-IIDness for each cluster for ablation studies.

Cluster	α	Num. of Clients
C1	1000	25
C2	1	25
C3	0.5	25
C4	0.1	25

We run two sets of experiments as shown in Table 1. The first set trains the global model from random initialization, starting with the cluster with the lowest degree of non-IIDness, C1, and hence only using 25% of the training set. We then incrementally add the other clusters into the pool and repeat the experiment. Each experiment is run 3 times using random seeds and the trained model is

used to evaluate both the initial and personalized accuracy on the entire test set. The two clusters with a lower degree of non-IIDness, C1 & C2, resulted in the highest initial accuracy and personalized accuracy for clients in C1. These accuracies fall when we include training data from clusters with a higher degree of non-IIDness, C3 & C4. This performance degradation motivates the line of work that aims to tackle the effects of data heterogeneity in single global FL. In contrast, the personalized accuracy for clients in C2, C3, & C4 improve as data within the same data heterogeneity regime is added to the training set.

We then ask if adding lower degrees of non-IIDness or even close to IID distributions would degrade the personalization accuracy for clients with greater degrees of non-IIDness? Similar to the first set of experiments, we start off with C4 instead and incrementally add more data from clients in C3, C2, & C1. Surprisingly, doing so improve both initial and personalized accuracy for all clusters, although it is worse in initial accuracy than the model trained with C1 & C2 only, 50% of the data. Based on this observation, we conclude that knowledge shared from lower degrees of non-IIDness distributions can improve the personalized accuracy for clients with higher degrees of non-IIDness distributions but the reverse is detrimental. Hence, we can leverage this insight to better control the impact of training with non-IID data on both the initial and personalized accuracy by clustering clients based on their degree of data heterogeneity.

5 CUAU: Weight Divergence of Client Updates

In Section 4, given the ground-truth degree of non-IIDness of each cluster, we show how varying degrees of data heterogeneity affects performance. However, in the FL setting, due to privacy, this information is not made public by the clients to the server. To this end, we propose *CUAU*, which is the weight divergence between the client update (*CU*) and the aggregated update (*AU*) of all clients in an FL round to estimate the degree of non-IIDness. In this section, we prove that the *CUAU* is bounded by the earth mover's distance (EMD) between the data distribution of the client and the population distribution and hence can be an accurate measure of data heterogeneity.

In Zhao et al. (2018), weight divergence is formulated as the difference of the aggregated weights of FL (w^{FL}) relative to weights optimized in a centralized manner ($w^{(cen)}$): $\|w^{FL} - w^{(cen)}\|/\|w^{(cen)}\|$. The authors showed that the EMD between the data distribution of the client and the population distribution is the root cause of this weight divergence through Proposition 3.1 in their paper. $w^{(cen)}$, however, is not available as the server does not have any client data to train on. Therefore, we build on their proposition shown below:

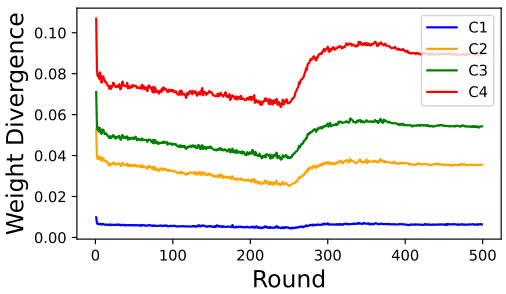

Figure 1: Mean *CUAU* of all clients in each cluster evaluated after every FL training round.

Proposition 1. *Given all N clients are selected for training, each with n_k samples following distribution p_k for client $k \in [\mathcal{K}]$. If $\nabla \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})$ is $\lambda_{\boldsymbol{x}|y=i}$-Lipchitz for each class $i \in [L]$, then we have the following in equality:*

$$CUAU_k = \|CU_{T,k} - AU_T\| \leq \frac{n_k}{n}\Big((a_k)^T \|w_{T-1} - w^{(cen)}_{T-1}\| + \eta \sum_{i=1}^{L} \|p_k(y=i) - p(y=i)\| \sum_{t=0}^{T} (a_k)^t g_{max}(w^{(cen)}_{ET-t})\Big) + Const. \tag{3}$$

where η stands for the learning rate; $Const.$ is the term independent of selected client k; $w^{(cen)}_{ET-t}$ is the weights trained in a centralized setting after E optimization steps; $g_{max}(\boldsymbol{w}) = \max_{i=1}^{L} \|\nabla \mathbb{E}_{\boldsymbol{x}|y=i} \log f_i(\boldsymbol{x}, \boldsymbol{w})\|$ and $a_k = 1 + \eta \frac{n_k}{n} \sum_{i=1}^{L} p_k(y=i) \lambda_{\boldsymbol{x}|y=i}$.

Detailed proof of Proposition 1 can be found in Appendix A. Based on Proposition 1, we have following remarks.

Remark 1. *$\|CU_{T,k} - AU\|$ can be seen as the weight divergence between each client update and the weight divergence induced by two parts. The first is the weight divergence after the $(T-1)$ round $\|w_{T-1} - w_{T-1}^{(cen)}\|$ and the second is the probability distance for the data distribution on client k compared with the actual distribution for the whole population, i.e., $\sum_{i=1}^{L} \|p_k(y=i) - p(y=i)\|$.*

Remark 2. *When all clients start from the same initialized weights, $\|p_k(y=i) - p(y=i)\|$ becomes the main cause of divergence, which is the earth mover's distance (EMD) of the data distribution of client k and the population data distribution.*

Based on Proposition 1, we validate that *CUAU* is a good metric to quantify the degree of heterogeneity of each client. To demonstrate this in practice, we adopt the same CIFAR10 cluster experimental setup presented in Section 1 and compute the *CUAU* of all clients after every FL training round. We then take the mean *CUAU* of all clients in each cluster and plot them in Figure 1. As the degree of non-IIDness increases from C1 to C4, the *CUAU* increases. Hence, we can accurately and efficiently estimate clusters using *CUAU* in a one-shot manner.

6 FedCUAU: Clustered FL using Weight Divergence

Motivated by our analysis (Section 4), we propose a clustered FL algorithm that clusters clients based on their degree of data heterogeneity, measured using *CUAU* (Section 5), with the goal of improving both the initial and personalized accuracy. A model is initialized for each cluster and knowledge is then shared among clusters in an unidirectional approach from the lowest to the highest degree of non-IIDness, *e.g.* client updates from C1 is used to update the weights of all other clusters as shown in Figure. 2. The initial accuracy is then computed using the model of the cluster with the lowest data heterogeneity and the personalized accuracy is obtained from each client's assigned cluster model.

6.1 Efficiency Considerations of FedCUAU

Recent works have shown that the deeper layers of the model are easily biased to the local distribution Luo et al. (2021); Oh et al. (2021); Zhuang et al. (2021); Ghosh et al. (2020). In other words, as the degree of non-IIDness increases, the feature similarity among local models decreases greatly in deeper layers. This is illustrated in Figure 3, where we group the ResNet18 layers into a few components: the first convolution layer (conv1), the ResNet blocks (RB1-4), and the last fully-connected (fc) layer. We then plot the mean cosine similarity between all pairs of local models in each cluster, using the setup defined in Section 4, for each component. Additionally, we show that there is a greater change in each component to fit the local data distribution as shown in Figure 3 (right), which shows the mean update magnitude of the weights of each component before and after fine-tuning.

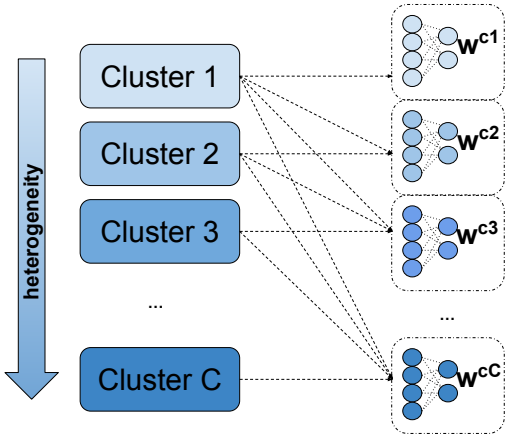

Figure 2: Knowledge shared among clusters.

Based on this observation, we make two modifications to reduce the cost of FedCUAU (Figure: 2) *1)* similar to IFCA Ghosh et al. (2020), we decouple the model and share the weights of earlier layers among clusters. We denote the shared weights as w^s and each cluster-specific weights as w^{Ci} where i is the cluster ID. *2)* We compute *CUAU* in a one-shot manner using a randomly initialized model based on the client and aggregated updates of the last fc layer. Hence, during the clustering stage, we can significantly reduce the communication cost from the server to each client by only sending the computation graph structure, along with its initialization definition and the fixed random seed. Additionally, each client only needs to send its updated fc layers back to the server.

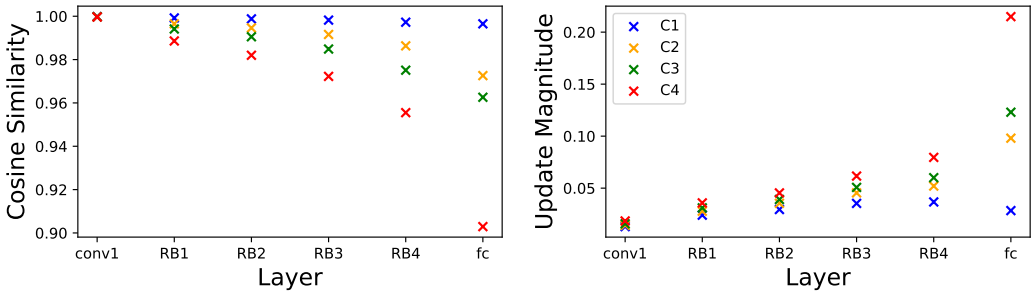

Figure 3: (left) Mean cosine similarity of representations between local models for each group of layers in each cluster. (right) Mean weight update magnitude before and after fine-tuning to the local data distribution in each cluster.

6.2 FEDCUAU ALGORITHM

Our proposed algorithm is shown in Algorithm 1. FedCUAU consists of two main stages. The first stage clusters the clients, in a one-shot manner, using *CUAU*, which is the weight divergence between each client's update (*CU*) and the aggregated update (*AU*). To this end, we initialize the model and sample all N clients in the client pool, performing a local update on all layers of the model for each client (lines 2-5,16-19). It is worth noting that the communication cost can be significantly reduced if the server sends the computation graph structure of the model, the initialization definition, and the seed instead of the model parameters themselves to all clients (Section 6.1). After which, we compute *AU* by taking the weighted mean of all *CU*s and use it to compute each client's *CUAU* (lines 6-8). This step can be efficiently calculated by just using the update of the last fc layer as described in Section 6.1. The clients are then clustered based on percentile points (line 9). Specifically, let $P_{i/\boldsymbol{\mathcal{C}}}$ (for $i = 1, ..., \boldsymbol{\mathcal{C}}$) be the ith percentile points of the set of *CUAU*, where $\boldsymbol{\mathcal{C}}$ is the total number of clusters defined, then client $k \in Ci$ if $P_{i-1/\boldsymbol{\mathcal{C}}} < CUAU_k \leq P_{i/\boldsymbol{\mathcal{C}}}$. The clusters are sorted in ascending order by data heterogeneity, with C1 representing the lowest degree of non-IIDness and C**C** representing the highest.

After the clients are clustered, the weights are reinitialized with its shared layers, w^s, and each cluster's layers, $w_{t,k}^{Ci}$ (line 10). For each of T training rounds, the server randomly sample K clients and sends the model parameters, along with each client's cluster ID (line 11-13). Each client then updates the shared layer and the last cluster's specific layer $w^{C\boldsymbol{\mathcal{C}}}$ for E mini-batches (line 21). Subsequently, each client freezes the shared layer and updates the cluster layers which ID is bigger or equal to his cluster ID as illustrated in Figure. 2 before sending the parameters back to the server (line 22-25). The server aggregates the weights and proceeds to the next round (line 14). Although FedAvg is used as an example to illustrate FedCUAU in Algorithm 1, other existing FL algorithms can also be easily adapted as shown in Section 6.3.

During testing, the initial global accuracy is computed using a single model, C1's model, and the personalized accuracy is computed using the client's assigned cluster model.

6.3 PERFORMANCE OF FEDCUAU

As our proposed FedCUAU focuses on cluster estimation and the information flow among clusters, it can be easily adapted and used to further boost the performance of existing FL algorithms. In this section, we adopted four popular FL algorithms: FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedBABU (Oh et al., 2021) and FedPer (Arivazhagan et al., 2019) and run them with and without FedCUAU on CIFAR-10 and FEMNIST.

Details of our CIFAR-10 and FEMNIST setup can be found in Section 4 and Caldas et al. (2018) respectively. We set the number of clusters $\boldsymbol{\mathcal{C}} = 2$ for all experiments in this section, where C1 represents the cluster with the lower degree of non-IIDness. Note that the clusters defined in this section is estimated using *CUAU* which differs from the clusters defined in Section 4 and Table 2, which is clustered using the ground-truth data heterogeneity. For CIFAR-10, we set the cluster-

Algorithm 1 **FedCUAU**: N is total number of clients. n is the total number of data samples; client k has n_k data samples. T is the total number of rounds. The number of local training steps is E and the number of clients participating in each round is K. w_t is the aggregated weights at round t. $\boldsymbol{\mathcal{C}}$ is the total number of clusters.

procedure FEDCUAU
  Initialize w_0 ▷ Initialize model without cluster-specific layers
  $b \leftarrow \{\}$
  for $k = 1, ..., N$ **do**
    $n_k, w_{0,k} \leftarrow$ CLIENTUPDATE(w_0)
  $AU \leftarrow \frac{1}{N} \sum_{k=1}^{N} \frac{n_k}{n} w_{0,k}$
  for $k = 1, ..., N$ **do**
    $CUAU_k \leftarrow \|CU_k - AU\|$
    if $P_{i-1/\boldsymbol{\mathcal{C}}} < CUAU_k \leq P_{i/\boldsymbol{\mathcal{C}}}$ **then** $b[k] \leftarrow i$ ▷ Cluster Assignment
  Initialize $w_0 \leftarrow \{w^s_{t,k}, w^{Ci}_{t,k}$ **for** $i, ..., \boldsymbol{\mathcal{C}}\}$ ▷ Initialize model with cluster-specific weights
  for $t = 1, ..., T$ **do**
    for all k in K **do** ▷ Randomly select K clients
      $n_k, w_{t,k} \leftarrow$ CLUSTERCLIENTUPDATE($w_{t,k}, b[k]$)
    $w_{t+1} = \frac{1}{K} \sum^{K} \frac{n_k}{\sum^{K} n_k} w_{t,k}$
  return w_T

procedure CLIENTUPDATE($w_{t,k}$)
  for $e = 1, ..., E$ **do**
    $w_{t,k} \leftarrow w_{t,k} - \eta \nabla f(w_{t,k})$
  return $n_k, w_{t,k}$

procedure CLUSTERCLIENTUPDATE($w_{t,k}, i$)
  $n_k, \{w^s_{t,k}, w^{C\boldsymbol{\mathcal{C}}}_{t,k}\} \leftarrow$ CLIENTUPDATE($\{w^s_{t,k}, w^{C\boldsymbol{\mathcal{C}}}_{t,k}\}$)
  for $j = i, ..., (\boldsymbol{\mathcal{C}} - 1)$ **do**
    for $e = 1, ..., E$ **do**
      $w^{Cj}_{t,k} \leftarrow \{w^s_{t,k}, w^{Cj}_{t,k}\} - \eta \nabla f(\{w^s_{t,k}, w^{Cj}_{t,k}\})$ ▷ Update cluster-specific weights
  return $n_k, w_{t,k}$

Table 3: Initial and personalized accuracy comparison for both Cifar10 and FEMNIST on various FL strategies. Averaged personalized accuracy is the fine-tuning accuracy over all clients in the client pool. FedCUAU boosts the performance of existing FL algorithms on both initial global accuracy and personalized accuracy. There is no initial accuracy for FedPer as it does not maintain a global model for evaluation. All experiments are repeated for 3 runs.

		Baseline		**+ FedCUAU**	
Dataset	**Algorithm**	**Init. Acc**	**Per. Acc (avg)**	**Init. Acc**	**Per. Acc (avg)**
CIFAR-10	FedAvg	82.49±0.30	86.85±0.26	83.75±0.08	87.14±0.32
	FedProx	82.39±0.16	86.56±0.30	83.00±0.32	86.84±0.29
	FedBABU	83.69±0.29	88.24±0.41	84.51±0.14	88.51±0.44
	FedPer	-	83.77±0.58	-	84.20±0.55
FEMNIST	FedAvg	78.62±0.43	74.73±0.13	80.12±0.56	78.54±0.08
	FedProx	79.51±1.30	75.18±0.19	80.70±0.85	78.75±0.17
	FedBABU	78.70±0.25	76.39±0.20	80.29±0.50	78.89±0.21
	FedPer	-	56.49±5.63	-	59.83±3.48

specific weights, w^{Ci}, for cluster i to be the last ResNet block (RB4) and the fc layer. Similarity, we set w^{Ci} to be the last convolution and fc layer for FEMNIST. Since the last fc layer is frozen in FedBABU and kept private in FedPer, we set w^{Ci} to be RB4 and the last convolution layer for CIFAR10 and FEMNIST respectively. Lastly, for FedProx, we use 0.001 for the hyperparameter of the regularization term.

Table 3 shows the results of FedCUAU in conjunction with the aforementioned FL algorithms on both the initial test accuracy and personalized test accuracy. All results are repeated for 3 separate runs. Note that there is no global model to evaluate the initial accuracy on for FedPer as the fc

Table 4: Initial comparison for both Cifar10 and FEMNIST on various FL strategies for both cluster weights; and personalized accuracy breakdown comparison for both initial and personalized accuracy for both cluster and cluster weights. There is no initial accuracy for FedPer as it does not maintain a global model for evaluation. All experiments are repeated for 3 runs.

Dataset	Algorithm	Cluster	Init. Acc	C1 Per. Acc	C2 Per. Acc	Per. Acc
CIFAR-10	FedAvg+FedCUAU	w^{C1}	83.75±0.08	84.70±0.28	87.00±0.20	85.85±0.24
		w^{C2}	82.49±0.30	84.12±0.15	89.59±0.36	86.85±0.26
	FedProx+FedCUAU	w^{C1}	83.00±0.32	84.26±0.41	86.88±0.13	85.57±0.27
		w^{C2}	82.39±0.16	83.70±0.42	89.42±0.17	86.56±0.30
	FedBABU+FedCUAU	w^{C1}	84.51±0.14	85.56±0.40	91.22±0.50	88.39±0.45
		w^{C2}	83.69±0.29	85.03±0.35	91.46±0.47	88.24±0.41
	FedPer+FedCUAU	w^{C1}	-	80.00±0.54	88.60± 0.34	84.30±0.22
		w^{C2}	-	79.15±0.70	88.39±0.57	83.77±0.58
FEMNIST	FedAvg+FedCUAU	w^{C1}	80.12±0.56	85.61±0.03	84.34±0.15	84.06±0.17
		w^{C2}	78.62±0.43	78.00±0.13	71.46±0.12	74.73±0.13
	FedProx+FedCUAU	w^{C1}	80.70±0.85	85.54±0.09	84.24±0.19	83.12±0.10
		w^{C2}	79.51±1.30	78.40±0.13	71.96±0.25	75.18±0.19
	FedBABU+FedCUAU	w^{C1}	80.29±0.50	85.76±0.16	82.35±0.18	82.80±0.17
		w^{C2}	78.70±0.25	78.44±0.14	72.02±0.26	76.39±0.20
	FedPer+FedCUAU	w^{C1}	-	66.68±1.61	60.26±3.30	63.47±2.45
		w^{C2}	-	60.00±5.92	53.00±5.34	56.49±5.63

layers are kept private on each client. As expected, FedCUAU improves existing FL algorithms by a considerable margin on both accuracy metrics. Initial accuracy of FedAvg improves by 1.53% & 1.82% for CIFAR10 & FEMNIST respectively, while personalized accuracy of FedAvg improves by 0.29% & 3.81%.

We investigate the gain seen in Table 3 by computing both the initial accuracy and personalized accuracy of both clusters using each cluster-specific weights as shown in Table 4. For CIFAR-10, each cluster's personalized accuracy is higher than the other for FedAvg, FedProx and FedBABU, verifying the effectiveness of clustering using FedCUAU. For FedPer, personalized accuracy is higher with w^{C1} for both clusters; learning a representation with less heterogeneous clients can better adapt with the private personalized layers, resulting from the fact that FedPer is already a personalized strategy with a client specific personalized layer, which means that the share part of the model can benefit more from the cluster weights trained by less heterogeneous clients.

Similarly, both initial and personalized accuracies are higher using C1's model for all FEMNIST experiments despite only using 50% of the total data to train w^{C1}. We hypothesize that this is due to the feature heterogeneity among clients as the handwriting differs among clients. Nonetheless, the significant improvement in C1's personalized accuracy resulted in an increase in the overall personalized accuracy shown in Table 3.

7 Conclusion

In this work, we focus on the clustering of clients in order to improve performance using existing FL algorithms. Through our experiments, we showed that data from lower data heterogeneity clusters could be used to improve the personalization of clients in higher data heterogeneity clusters. Hence, by restricting the knowledge shared from higher to lower data heterogeneity clusters, we mitigate the performance degradation caused by non-IID data on both the initial accuracy and the personalization of lower data heterogeneity clusters. We also showed that using the weight divergence between each client update and the aggregated update of all clients in an FL round is an effective and efficient measure of heterogeneity and leveraged it in our one-shot clustering algorithm. Despite the gains of FedCUAU, we are still facing with other challenges in FL that might limit FedCUAU's applicability, *e.g.* quantity skew or feature distribution shift. Hence, a possible avenue for future work would be to explore how clustering can tackle the other areas of data heterogeneity.

REFERENCES

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019.

Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Daoyuan Chen, Dawei Gao, Weirui Kuang, Yaliang Li, and Bolin Ding. pfl-bench: A comprehensive benchmark for personalized federated learning. *Advances in Neural Information Processing Systems*, 2022.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning, 2021.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*. PMLR, 2021.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 2017.

Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust rederated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models, 2021.

Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos I Venieris, and Nicholas D Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *arXiv preprint arXiv:2102.13451*, 2021.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*. PMLR, 2021b.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 2021.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 2021.

Koji Matsuda, Yuya Sasaki, Chuan Xiao, and Makoto Onizuka. Fedme: Federated learning via model exchange. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 2022a.

Koji Matsuda, Yuya Sasaki, Chuan Xiao, and Makoto Onizuka. An empirical study of personalized federated learning. *arXiv preprint arXiv:2206.13190*, 2022b.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

Xinchi Qiu, Javier Fernandez-Marques, Pedro PB Gusmao, Yan Gao, Titouan Parcollet, and Nicholas Donald Lane. Zerofl: Efficient on-device training for federated learning with local sparsity. *arXiv preprint arXiv:2208.02507*, 2022.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pp. 7252–7261. PMLR, 2019.

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. Collaborative unsupervised visual representation learning from decentralized data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

A PROOF OF PROPOSITION 1

Proof. Based on the definition, assume that each client perform E local optimization steps, $w_{T,k}$ is the weights after local training at communication round T at client k. We also write the $w_{ET}^{(cen)}$ as the weight after ET steps of optimization in centralized manner.

Based on definition of CU_k and AU, we have:

$$\begin{aligned}\|CU_{T,k} - AU_T\| &= \|w_{T,k} - w_T\| \\ &= \|(w_{T,k} - w^{(cen)}) - (w_T - w^{(cen)})\| \\ &= \|(w_{T,k} - w^{(cen)}) - (\frac{1}{N}\sum_{i=1}^{N} w_{T,i} - w^{(cen)})\| \\ &\leq \|(w_{T,k} - w^{(cen)})\| + \|(\frac{1}{N}\sum_{i=1}^{N} w_{T,i} - w^{(cen)})\|\end{aligned}$$

Based on Proposition 3.1 in Zhao et al. (2018), assuming that only 1 is selected, the first part of the right hand side can be bounded by:

$$\begin{aligned}\|(w_{T,k} - w^{(cen)})\| \leq \frac{n_k}{n}(a_k)^T\|w_{T-1} - w_{T-1}^{(cen)}\| \\ + \eta\frac{n_k}{n}\sum_{i=1}^{L}\|p_k(y=i) - p(y=i)\|\sum_{t=1}^{T-1}(a_k)^t g_{max}(w_{ET-t-1}^{(cen)})\end{aligned}$$

where $g_{max}(\mathbf{w}) = \max_{i=1}^{L}\|\nabla\mathbb{E}_{\mathbf{x}|y=i}\log f_i(\mathbf{x},\mathbf{w})\|$ and $a_k = 1 + \eta\sum_{i=1}^{L} p_k(y=i)\lambda_{\mathbf{x}|y=i}$.

Also use the same proposition, assuming that all N clients are selected for training, then the second part of the right hand side can be bounded by:

$$\begin{aligned}\|(\frac{1}{N}\sum_{i=1}^{N} w_{T,i} - w^{(cen)})\| \leq \sum_{k=1}^{N}\frac{n_k}{n}\Big((a_k)^T\|w_{T-1} - w_{T-1}^{(cen)}\| \\ + \eta\sum_{i=1}^{L}\|p_k(y=i) - p(y=i)\|\sum_{t=1}^{T-1}(a_k)^t g_{max}(w_{ET-t-1}^{(cen)})\Big) = Const.\end{aligned}$$

Since it is summed over all clients, it is independent of the particular client. Hence, it can be written as $Const.$ in Proposition 1. □