Efficient Distribution Similarity Identification in Clustered Federated Learning via Principal Angles Between Client Data Subspaces

Saeed Vahidian*[1], Mahdi Morafah*[1], Weijia Wang[1], Vyacheslav Kungurtsev[2], Chen Chen[3], Mubarak Shah[3], Bill Lin[1]

[1]UC San Diego
[2]Czech Technical University
[3]UCF

{svahidia, mmorafah, wweijia, billlin}@ucsd.com, kunguvya@fel.cvut.cz, {chen.chen, shah}@crcv.ucf.edu

Abstract

Clustered federated learning (FL) has been shown to produce promising results by grouping clients into clusters. This is especially effective in scenarios where separate groups of clients have significant differences in the distributions of their local data. Existing clustered FL algorithms are essentially trying to group together clients with similar distributions so that clients in the same cluster can leverage each other's data to better perform federated learning. However, prior clustered FL algorithms attempt to learn these distribution similarities indirectly during training, which can be quite time consuming as many rounds of federated learning may be required until the formation of clusters is stabilized. In this paper, we propose a new approach to federated learning that directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces. Each client applies a truncated singular value decomposition (SVD) step on its local data in a single-shot manner to derive a small set of principal vectors, which provides a signature that succinctly captures the main characteristics of the underlying distribution. This small set of principal vectors is provided to the server so that the server can directly identify distribution similarities among the clients to form clusters. This is achieved by comparing the similarities of the principal angles between the client data subspaces spanned by those principal vectors. The approach provides a simple, yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews. Our clustered FL approach also enables convergence guarantees for non-convex objectives.

Introduction

Federated Learning (FL) (McMahan and Ramage 2017) enables a set of clients to collaboratively learn a shared prediction model without sharing their local data. Some FL approaches aim to train a common global model for all clients (McMahan et al. 2017; Li et al. 2020; Wang et al. 2020; Karimireddy et al. 2020; Mendieta et al. 2022). However, in many FL applications where there may be data heterogeneity among clients, a single relevant global model may not exist. Alternatively, personalized FL approaches have been studied. One approach is to first train a global model and then allow each client to fine-tune it via a few rounds of stochastic gradient descent (SGD) (Fallah, Mokhtari, and Ozdaglar 2020; Vahidian et al. 2022; Liang et al. 2020a). Another approach is for each client to jointly train a global model as well as a local model, and then interpolate them to derive a personalized model (Deng, Kamani, and Mahdavi 2020; Mansour et al. 2020). In the former case, the approach often fails to derive a model that generalizes well to the local distributions of each client. In the latter case, when local distributions and the average distribution are far apart, the approach often degenerates to every client learning only on its own local data. Recently, clustered FL (Ghosh et al. 2020; Sattler, Müller, and Samek 2021; Morafah et al. 2022; Mansour et al. 2020) has been proposed to allow the grouping of clients into clusters so that clients belonging to the same cluster can share the same optimal model. Clustered FL has been shown to produce significantly better results, especially when separate groups of clients have significant differences in the distributions of their local data. This possibly due to distinct learning tasks or the mixture of distributions of the local data considered, not necessarily limited to simpler forms of data heterogeneity such as label skews from otherwise the same dataset.

Essentially, what prior clustered FL algorithms are trying to do is to group together clients with similar distributions so that clients in the same cluster can leverage each other's data to perform federated learning more effectively. Previous clustered FL algorithms attempt to *learn* these distribution similarities *indirectly* when clients learn the cluster to which they should belong as well as the cluster model during training. For example, the clustered FL approach presented in (Sattler, Müller, and Samek 2021) alternately estimates the cluster identities of clients and optimizes the cluster model parameters via SGD.

Unfortunately, the prior clustered FL approaches have the following *challenges* which in turn limits their applicability in real-world problems. 1) Since previous clustered FL training algorithms start with randomly initialized cluster models that are inherently noisy, the overall training process can be quite time consuming as many rounds of federated learning may be required until the formation of clusters is stabilized. 2) Approaches like IFCA (Ghosh et al. 2020) assumes a

*These authors contributed equally.

pre-defined number of clusters, but requiring the number of clusters to be fixed *a priori*, regardless of the differences in the actual data distributions or learning tasks among the clients, could lead to poor performance for many clients. 3) In each iteration, all cluster models have to be downloaded by the active clients in that round, which can be very costly in communications. 4) Both of the approaches i.e., those that train a common global model for all clients and personalized approaches including IFCA lack the flexibility to trade off between *personalization* and *globalization*. The above-mentioned drawbacks of the prior works, naturally lead to the following important question. *How a server can realize clustered FL efficiently by grouping the clients into clusters in a one-shot manner without requiring the number of clusters to be known apriori, but with substantially less communication cost?* In this work, we propose a novel algorithm, Principal Angles analysis for Clustered Federated Learning (PACFL), to address the above-mentioned challenges of clustered FL.

Our contributions. We propose a new algorithm, PACFL, for federated learning that *directly* aims to efficiently identify distribution similarities among clients by analyzing the *principal angles between the client data subspaces*. Each client wishing to join the federation applies a truncated SVD step on its local data in a *one-shot* manner to derive a *small set of principal vectors*, which form the principal bases of the underlying data. These principal bases provide a *signature* that succinctly captures the main characteristics of the underlying distribution. The client then provides this small set of principal vectors to the server so that the server can directly identify distribution similarities among the clients to form clusters. The privacy of data is preserved since no client data is ever sent to the server but a few (2-5) principal vectors out of ≈ 500. Thus, the clients data cannot be reconstructed from those (2-5) number of left singular vectors. However, in privacy sensitive setups to provide extra protection and prevent any information leakage from clients to server, mechanisms like the ones presented in (Bonawitz et al. 2017), or encryption mechanism or differential privacy method that achieves this end can be employed.

On the server side, it efficiently identifies distribution similarities among clients by comparing the principal angles between the client data subspaces spanned by the provided principal vectors – the greater the difference in data heterogeneity between two clients, the more orthogonal their subspaces. Unlike prior clustered FL approaches, which require time consuming iterative learning of the clusters and substantial communication costs, our approach provides a simple yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews. Clients can immediately collaborate with other clients in the same cluster from the get go.

Our novel PACFL approach has the flexibility to trade off between *personalization* and *globalization*. PACFL can naturally span the spectrum of identifying IID data distribution scenarios in which all clients should share training within only 1 cluster, to the other end of the spectrum where clients have extremely Non-IID data distributions in which each client would be best trained on just its own local data (i.e., each client becomes its cluster).

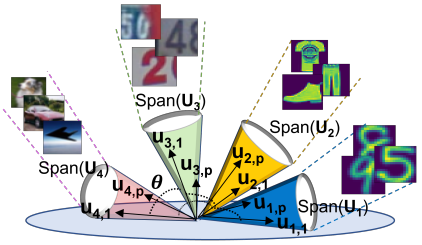


Figure 1: There must be a translation protocol enabling the server to understand the similarity/dissimilarity of the clients' data without sharing data. This 2D figure intuitively demonstrates how the principal angle between the client data subspaces captures the statistical heterogeneity. Fig. 1 Shows the subspaces spanned by the $\mathbf{U}_p$s of four different datasets (left to right: CIFAR-10, SVHN, FMNIST, and USPS). As can be seen the principal angle between the subspaces of CIFAR-10 and SVHN is smaller than that of CIFAR-10 and USPS.

Our framework also naturally provides an elegant approach to handle newcomer clients unseen at training time by matching them with a cluster model that the client can further personalized with local training. Realistically, new clients may arrive to the federation after the distributed training procedure. In our framework, the newcomer client simply provides its principal vectors to the server, and the server identifies via angle similarity analysis which existing cluster model would be most suitable, or the server can inform the client that it should train on its own local data to form a new cluster if the client's data distribution is not sufficiently similar to the distributions of the existing clusters. On the other hand, it is generally unclear how prior personalized or clustered FL algorithms can be extended to provide newcomer clients with similar capabilities.

Finally, we provide a convergence analysis of PACFL in the supplementary material.

Clustered Federated Learning

Preliminaries

Principal angles between two subspaces. Let $\mathcal{U} = \text{span}\{\mathbf{u}_1, ..., \mathbf{u}_\text{p}\}$ and $\mathcal{W} = \text{span}\{\mathbf{w}_1, ..., \mathbf{w}_\text{q}\}$ be p and q-dimensional subspaces of $\mathbf{R^n}$ where $\{\mathbf{u}_1, ..., \mathbf{u}_p\}$ and $\{\mathbf{w}_1, ..., \mathbf{w}_q\}$ are orthonormal, with $1 \leq p \leq q$. There exists a sequence of p angles $0 \leq \Theta_1 \leq \Theta_2 \leq ... \leq \Theta_p \leq \pi/2$ called the principal angles, which are defined as

$$\Theta\left(\mathcal{U}, \mathcal{W}\right) = \min_{\mathbf{u}\in\mathcal{U}, \mathbf{w}\in\mathcal{W}} \arccos\left(\frac{\left|\mathbf{u}^T\mathbf{w}\right|}{\|\mathbf{u}\|\,\|\mathbf{w}\|}\right) \tag{1}$$

where $\|.\|$ is the induced norm. The smallest principal angle is $\Theta_1\left(\mathbf{u}_1, \mathbf{w}_1\right)$ with vectors $\mathbf{u}_1$ and $\mathbf{w}_1$ being the corresponding principal vectors. The principal angle distance is a metric for measuring the distance between subspaces (Jain, Netrapalli, and Sanghavi 2013). Additional background is presented in the supplementary material.

Dataset	CIFAR-10	SVHN	FMNIST	USPS
CIFAR-10	0 (0)	6.13 (12.3)	45.79 (91.6)	66.26 (132.5)
SVHN	6.13 (12.3)	0 (0)	43.42 (86.8)	64.86 (129.7)
FMNIST	45.79 (91.6)	43.42 (86.8)	0 (0)	43.36 (86.7)
USPS	66.26 (132.5)	64.86 (129.7)	43.36 (86.7)	0 (0)

Table 1: This table shows how distribution similarities between datasets can be accurately estimated by a proximity matrix of principal angles. Entries are $x(y)$, where x and y are respectively the smallest principal angle and summation over the principal angles between two datasets. p in $\mathbf{U}_p$ is 2.

How Principal Angles Can Capture the Similarity Between Data/Features

The *cosine similarity* is a distance metric between vectors that is known to be more tractable and interpretable than the alternatives while exhibiting high precision clustering properties, see, e.g. (Qian et al. 2004). The idea is to note that for any two vectors x and y, by the dot product calculation $x \cdot y = \|x\|\|y\| \cos\theta$, we see that inverting the operation to solve for θ yields the angle between two vectors as emanating from the origin, which is a scale-invariant indication of their alignment. It presents a natural geometric understanding of the proportional volume of the embedded space that lies between the two vectors. Finally, by choosing to cluster using the data rather than the model, the variance of each SGD sample declines resulting in smoother training. By contrast, clustering by model parameters has the effect of increasing the bias of the clients' models to be closer to each other.

In order to obtain a computationally tractable small set of vectors to represent the data features, we propose to apply truncated SVD on each dataset. We take a small set of principal vectors, which form the principal bases of the underlying data distribution. Truncated SVD is known to yield a good quality balance between computational expense and representative quality of representative subspace methods (Talwalkar et al. 2013). Assume there are K number of datasets. We propose to apply truncated SVD (detailed in the supplementary material) on these data matrices, $\mathbf{D}_k, k = 1, ..., K$, whose columns are the input features of each dataset. Further, let $\mathbf{U}_p^k = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p]$, $(p \ll \text{rank}\,(\mathbf{D}_\text{k}))$ be the p most significant left singular vectors for dataset k. We constitute the proximity matrix $\mathbf{A}$ as in Eq. 2 whose entries are the smallest principle angle between the pairs of $\mathbf{U}_p^k$ or as in Eq. 3 whose entries are the summation over the angle in between of the corresponding $\mathbf{u}$ vectors (in identical order) in each pair within $\mathbf{U}_p^k$, where $\mathbf{tr}\,(.)$ is the trace operator, and

$$\mathbf{A}_{i,j} = \Theta_1\left(\mathbf{U}_p^i, \mathbf{U}_p^j\right), \quad i, j = 1, ..., K \tag{2}$$

$$\mathbf{A}_{i,j} = \mathbf{tr}\left(\arccos\left(\mathbf{U}_p^{i\,T} * \mathbf{U}_p^j\right)\right), \quad i, j = 1, ..., K \tag{3}$$

The smaller the entry of $\mathbf{A}_{i,j}$ is, the more similar datasets i and j are [1]. Before we proceed further, through some experiments on benchmark datasets, we highlight how the proposed method perfectly distinguishes different datasets based on their hidden data distribution by inspecting the angle between their data subspaces spanned by their first p left singular vectors. For a visual illustration of the result, we refer to Fig. 1. As can be seen the principal angle between the subspaces of CIFAR-10 and SVHN is smaller than that of CIFAR-10 and USPS. Table 1 shows the exact principal angles between every pairs of these datasets' subspaces. The entries of this table is presented as $x(y)$, where x is the smallest principal angle between two datasets obtained from Eq. 2, and y is the summation over the principal angles between two datasets obtained from Eq. 3. Table 1 reveals that the similarity and dissimilarity of the four different datasets have been accurately captured by the proposed method. We will provide more examples in the supplementary material and will show that the similarity/dissimilarity being captured by the proposed method is consistent with well-known distance measures between two distributions including Bhattacharyya Distance (BD), Maximum Mean Discrepancy (MMD) (Gretton et al. 2012), and Kullback–Leibler (KL) distance (Hershey and Olsen 2007).

Overview of PACFL

In this section, we begin by presenting our PACFL framework. The proposed approach, PACFL, is described in Algorithm 1. We first turn our attention to clustering clients data in a federated network. The proposed method is one-shot clustering and can be used as a simple pre-processing stage to characterize personalized federated learning to achieve superior performance relative to the recent iterative approach for clustered FL proposed in (Ghosh et al. 2020). Before federation, each available client, k, performs truncated SVD on its own data matrix, $\mathbf{D}_k$ [2], and sends the p most significant left singular vectors $\mathbf{U}_p$, as their data *signature* to the central server. Next, the server obtains the proximity matrix $\mathbf{A}$ as in Eq. 2 or Eq. 3 where $K = |\mathcal{S}_t|$, and $\mathcal{S}_t$ is the set of available clients. When the number of clusters is unknown, for forming disjoint clusters, the server can employ agglomerative hierarchical clustering (HC) (Day and Edelsbrunner 1984) on the proximity matrix $\mathbf{A}$. *For more details on HC, please see the supplementary material.* Hence, the cluster ID of clients is determined.

For training, the algorithm starts with a single initial model parameters θ_g^0. In the first iteration of PACFL a random subset of available clients $\mathcal{S}_t \subseteq [N]$, $|\mathcal{S}_t| = n$ is selected by the server and the server broadcasts θ_g^0 to all clients. The clients start training on their local data and perform some steps of stochastic gradient descent (SGD) updates, and

[1] It is noteworthy that in practice both of these equations work accurately. However, theoretically and rigorously speaking, when the number of the principal vectors, p, is bigger than 1, it can happen that one of the principal vectors of client k yields a small angle with its corresponding one for client k' while the other principal vectors of client k yield big angle with their corresponding ones for client k'. With that in mind, Eq. 3 is a more rigorous measure and it always truly captures the similarity between the client data subspaces.

[2] Considering a client owns M data samples, each including N features, we assumed that the M data samples are organized as the columns of a matrix $D_k \in R^{N \times M}$.

Algorithm 1: PACFL

Require: Number of available clients N, sampling rate $R \in (0, 1]$, clustering threshold β
Server: Initialize the server model with θ_g^0.
for each round t = 1, 2,... **do**
  $m \leftarrow \max(R \cdot N, 1)$
  $\mathcal{S}_t \leftarrow \{k_1, ..., k_n\}$ % set of n available clients %
  for each client $k \in \mathcal{S}_t$ **in parallel do**
    if $t = 1$ % It is done in one-shot % **then**
      client k sends $\mathbf{U}_p^k$ to the server
      $\mathbf{U} = [\mathbf{U}_p^{k_1}, ..., \mathbf{U}_p^{k_n}]$
      $\mathbf{A} \leftarrow$ server forms $\mathbf{A}$ based on Eq. 2 or Eq. 3
      $\{C_1, ..., C_Z\} = \mathrm{HC}(\mathbf{A}, \beta)$
      $\theta_{g,z}^0 \leftarrow \theta_g^0$ % initializing all clusters with θ_g^0 %
    else if k is a new arriving client **then**
      client k sends $\mathbf{U}_p^k$ to the server
      $\mathbf{A}$, $\mathbf{U}$ = PME($\mathbf{A}$, $\mathbf{U}$, $\mathbf{U}_p^k$) % Alg. 2 %
      $\{C_1, ..., C_Z\} \leftarrow \mathrm{HC}(\mathbf{A}, \beta)$ % Update the clusters, determine the cluster ID of new client k %
      client k receives the corresponding cluster model $\theta_{g,z}^t$ from the server
    else
      client k sends its cluster ID to the server and receives the corresponding cluster model $\theta_{g,z}^t$ from the server
    end if
    $\theta_{k,z}^{t+1} \leftarrow \mathrm{ClientUpdate}(k; \theta_{g,z}^t)$: by SGD training
  end for
  for $z = 1 : Z$ % Z is the number of formed clusters % **do**
    $\theta_{g,z}^{t+1} = \sum_{k \in C_z} |D_k| \theta_{k,z}^{t+1} / \sum_{k \in C_z} |D_k|$ % model averaging for each cluster %
  end for
end for

Algorithm 2: Proximity Matrix Extension (PME)

Input: $\mathbf{A}_{old}$ % $M \times M$ proximity matrix formed by M number of seen clients %
Input: $\mathbf{U}_{old} = [U_p^1, ..., U_p^M]$ % Set of first p significant singular vectors of the M seen clients %
Input: $\mathbf{U}_{new} = [U_p^1, ..., U_p^B]$ % The set of first p significant singular vectors of the B new clients %
Output: $\mathbf{A}_{extended}$ % The extended $(M+B)\times(M+B)$ proximity matrix %
Output: $\mathbf{U}_{extended}$
function PME($\mathbf{A}_{old}$, $\mathbf{U}_{old}$, $\mathbf{U}_{new}$)
  $\mathbf{A}_{extended} \leftarrow [\mathbf{0}]_{(M+B)\times(M+B)}$
  $U_{extended} \leftarrow [\mathbf{U}_{old}, \mathbf{U}_{new}]$
  $\mathbf{A}_{extended}[1 : M, 1 : M] = \mathbf{A}_{old}[:, :]$
  $\mathbf{A}_{extended}[M : M + B, M : M + B]$ % can be calculated based on Eq. 2 or Eq. 3 %
  Return $\mathbf{A}_{extended}$, $\mathbf{U}_{extended}$
end function

get the updated model. The clients will only need to send their cluster membership ID and model parameters back to the central server. After receiving the model and cluster ID memberships from all the participating clients, the server then collects all the parameter updates from clients whose cluster ID are the same and conducts model averaging within each cluster. It is noteworthy that in Algorithm 1, β stands for the Euclidean distance between two clusters and is a parameter in HC.

Desirable properties of PACFL. Unlike prior work on clustered federated learning (Ghosh et al. 2020; Sattler, Müller, and Samek 2021), PACFL has much greater flexibility in the following sense. First, from a *practical* perspective, one of the desirable properties of PACFL is that it can handle partial participation of clients. In addition, PACFL does not require to know in advance whether certain clients are available for participation in the federation. Clients can join and leave the network abruptly. In our proposed approach, the new clients that join the federation just need to send their data signature to the server and the server can easily determine the cluster IDs of the new clients by constituting a new proximity matrix without altering the cluster IDs of the other clients. In PACFL, the prior information about the availability of certain clients is not required.

Second, PACFL can form the best fitting number of clusters, if a fixed number of clusters is not specified. However, in IFCA (Ghosh et al. 2020), the number of clusters has to be known apriori. Third, one-shot client clustering can be placed by PACFL for the available clients before the federation and the prior information about the availability and the number of certain clients is not required. In contrast, IFCA constructs the clusters iteratively by alternating between cluster identification estimation and loss function minimization which is costly in communication.

Fourth, PACFL does not add significant additional computational overhead to the FedAvg baseline algorithm as it only requires running one-shot HC clustering before training. With that in mind, the computational complexity of the PACFL algorithm is the same as that of FedAvg plus the computational complexity of HC in one-shot ($(O(N^2))$ where N is the total number of clients).

Fifth, in case of either a certain and a fixed number of clients are not available at the initial stage or clients join and leave the network abruptly, clustering by PACFL can easily be applied in a few stages as outlined in Algorithm 2. Each new clients that become available for the federation, sends the signature of its data to the server and the server aggregates the signature of existing clients and the new ones as in $U_{extended}$ (Algorithm 2). Next, the server obtains the proximity matrix $\mathbf{A}$ as in Eq. 3 or Eq. 4 where all the new clients included. *By keeping the same distance threshold* as before, the cluster ID of the new clients are determined without changing the cluster ID of the old clients.

Sixth, we should note that we tried some other clustering methods including graph clustering methods (Hallac, Leskovec, and Boyd 2015; Sarcheshmehpour, Leinonen, and Jung 2021) for PACFL and we noticed that the clustering

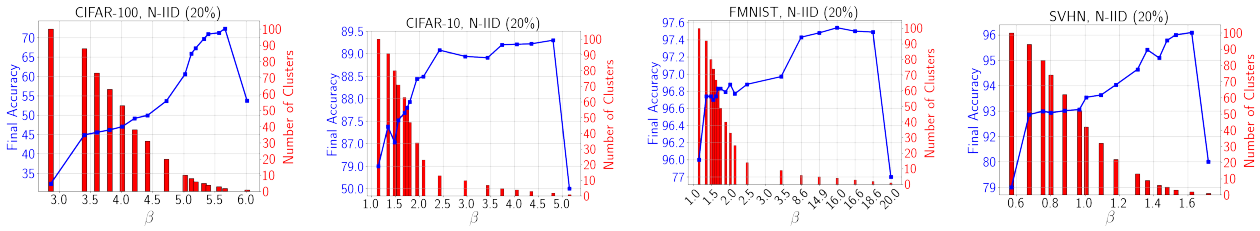


Figure 2: Test accuracy performance of PACFL versus the clustering threshold β (when the proximity matrix obtained as in Eq. 2), and the number of fitting clusters for Non-IID label skew (20%) on CIFAR-10/100, FMNIST, and SVHN datasets. Each point in the plots are obtained by 200 communication rounds with local epoch of 10, local batch size of 10 and SGD local optimizer.

algorithm does not play a crucial role in PACFL. As long as the clustering algorithm itself does not require the number of clusters to be known in advance, it can be applied in PACFL.

Seventh, the server cannot make use of the well known distribution similarity measures such as BD, MMD (Gretton et al. 2012), and KL (Hershey and Olsen 2007) to group the clients into clusters due to the privacy constraints as they require accessing the data or important moments of the data distributions. As shown in Fig. 1, and Table 1 and also as will be shown in the Experiments Section, the proposed approach presents a simple and alternative solution to the above-mentioned measures in FL setups.

We also provide a convergence analysis of the method in the supplementary material. The framework we use is from (Haddadpour and Mahdavi 2019), which considers nonconvex learning with Non-IID data. Indeed unlike other works we can obtain guarantees for nonconvex objectives, as appropriate for deep learning, because the clustering is performed on the *data* and not the parameters, thus no longer suffering from associated issues of multi-modality (multiple separate local minima). We shall see that it recovers the state-of-the-art (SOTA) convergence rate and performance guarantees for Non-IID data.

Experiments

Datasets and Models. We use image classification task and 4 popular datasets, i.e., FMNIST (Xiao, Rasul, and Vollgraf 2017), SVHN (Netzer et al. 2011), CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), to evaluate our method. For all experiments, we consider LeNet-5 (LeCun et al. 1989) architecture for FMNIST, SVHN, and CIFAR-10 datasets and ResNet-9 (He et al. 2016) architecture for CIFAR-100 dataset.

Baselines and Implementation. To assess the performance of the proposed method against the SOTA, we compare PACFL against the following set of baselines. For baselines that train a single global model across all clients, we compare with FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020) FedNova (Wang et al. 2020), and SCAFFOLD (Karimireddy et al. 2020). For SOTA personalized FL methods, the baselines include LG-FedAvg (Liang et al. 2020b), Per-FedAvg (Fallah, Mokhtari, and Ozdaglar 2020), Clustered-FL (CFL) (Sattler, Müller, and Samek 2021), and IFCA (Ghosh et al. 2020). In all experiments, we assume 100 clients are available and 10% of them are sampled randomly at each round. Unless stated otherwise, throughout the experiments, the number of communication rounds is 200 and each client performs 10 locals epochs with batch size of 10 and local optimizer is SGD. We let p in $\mathbf{U}_p$ be 3-5. Please refer to the supplementary material for more details about the experimental setup.

Overall Performance

We compare PACFL with all the mentioned SOTA baselines for two different widely used Non-IID settings, i.e. Non-IID label skew, and Non-IID Dirichlet label skew (Li et al. 2021a). We present the results of Non-IID label skew in the main paper and that of the Non-IID Dirichlet label skew in the supplementary material. We report the mean and standard deviation for the average of final local test accuracy across all clients over 3 runs.

Non-IID Label Skew. In this setting, we first randomly assign $\varrho\%$ of the total available labels of a dataset to each client and then randomly distribute the samples of each label amongst clients own those labels as in (Li et al. 2021b). In our experiments we use Non-IID label skew 20%, and 30%, i.e. $\varrho = \{20, 30\}\%$ respectively. Table 2 shows the results for Non-IID label skew 20%. We report the results of Non-IID label skew 30% in the supplementary material. As can be seen, global FL baselines, i.e. FedAvg, FedProx, FedNova, and SCAFFOLD perform very poorly. That's due to weight divergence and model drift issues under heterogeneous setting (Zhao et al. 2018). We can observe from Table 2 that PACFL consistently outperforms all SOTA on all datasets. In particular, focusing on CIFAR-100, PACFL outperforms all SOTA methods (by $+19\%, +18\%, +19\%, +18\%$ for FedAvg, FedProx, FedNova, SCAFFOLD) as well as all the personalized competitors (by $+27\%, +13\%, +1.5\%, +33\%$ for LG, PerFedAvg, IFCA, CFL). We tuned the hyperparameters in each baseline to obtain the best results. IFCA achieved the best performance with 2 clusters which is consistent with the results in (Ghosh et al. 2020).

Algorithm	FMNIST	CIFAR-10	CIFAR-100	SVHN
SOLO	95.92 ± 0.57	79.22 ± 1.67	32.28 ± 0.23	79.72 ± 1.37
FedAvg	77.3 ± 4.9	49.8 ± 3.3	53.73 ± 0.50	80.2 ± 0.8
FedProx	74.9 ± 2.6	50.7 ± 1.7	54.35 ± 0.84	79.3 ± 0.9
FedNova	70.4 ± 5.1	46.5 ± 3.5	53.61 ± 0.42	75.4 ± 4.8
Scafold	42.8 ± 28.7	49.1 ± 1.7	54.15 ± 0.42	62.7 ± 11.6
LG	96.80 ± 0.51	86.31 ± 0.82	45.98 ± 0.34	92.61 ± 0.45
PerFedAvg	95.95 ± 1.15	85.46 ± 0.56	60.19 ± 0.15	93.32 ± 2.05
IFCA	97.15 ± 0.01	87.99 ± 0.15	71.84 ± 0.23	95.42 ± 0.06
CFL	77.93 ± 2.19	51.11 ± 1.01	40.29 ± 2.23	73.62 ± 1.76
PACFL	$\mathbf{97.54 \pm 0.08}$	$\mathbf{89.30 \pm 0.41}$	$\mathbf{73.10 \pm 0.21}$	$\mathbf{95.77 \pm 0.18}$

Table 2: Test accuracy comparison across different datasets for Non-IID label skew (20%). For each baseline, the average of final local test accuracy over all clients is reported. We run each baseline 3 times for 200 communication rounds with local epoch of 10.

Globalization and Personalization Trade-off

To cope with the statistical heterogeneity, previous works incorporated a proximal term in local optimization or modified the model aggregation scheme at the server side to take the advantage of a certain level of personalization (Li et al. 2020; Vahidian, Morafah, and Lin 2021; Deng, Kamani, and Mahdavi 2020). Though effective, they lack the flexibility to trade off between *personalization* and *globalization*. Our proposed PACFL approach can naturally provide this globalization and personalization trade-off. Fig. 2 visualizes the accuracy performance behavior of PACFL versus different values of β which is the L_2 (Euclidean) distance between two clusters when the proximity matrix obtained as in Eq. 2, or Eq. 3. In other words, β is a threshold controlling the number of clusters as well as the similarity of the data distribution of clients within a cluster under Non-IID label skew. The blue curve and the red bars demonstrate the accuracy, and the number of clusters respectively for each β. Varying β in a range which depends upon the dataset, PACFL can sweep from training a fully global model (with only 1 cluster) to training fully personalized models for each client.

As is evident from Fig. 2, the behaviour of PACFL on each dataset is similar. In particular, increasing β, decreases the number of clusters (by grouping more number of clients within each cluster and sharing more training) which realizes more globalization. When β is big enough, PACFL will group all clients into 1 cluster and the scenario reduces to the FedAvg baseline (pure globalization). On the contrary, decreasing β, increases the number of clusters, which leads to more personalization. When β is small enough, individual clusters would be formed for each client and the scenario degenerates to the SOLO baseline (pure personalization). As demonstrated, on all datasets, all clients benefit from some level of globalization. This is the reason why decreasing the number of clusters can improve the accuracy performance in comparison to SOLO. In general, *finding the optimal trade-off between globalization and personalization depends on the level of heterogeneity of tasks, the intra-class distance of the dataset, as well as the data partitioning across the clients*. This is precisely what PACFL is designed to do, to find this optimal trade-off before initiating the federation via the proximity matrix at the server. IFCA lacks this trade-off capability as it must define a fixed number of clusters $(C > 1)$ or with $C = 1$ it would degenerate to FedAvg.

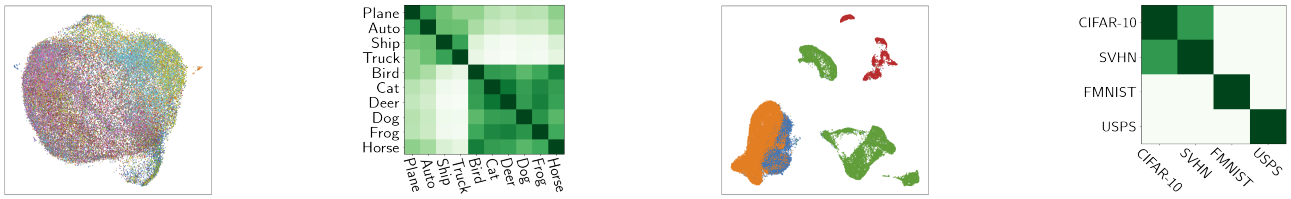
Mixture of 4 Datasets

Existing studies have been evaluated on simple partitioning strategies, i.e., Non-IID label skew (20%) and (30%) (Li et al. 2021b). While these partitioning strategies synthetically simulate Non-IID data distributions in FL by partitioning a dataset into multiple smaller Non-IID subsets, they cannot design real and challenging Non-IID data distributions. According to the prior sections, due to the small intra-class distance (similarity between distribution of the classes) in the used benchmark datasets, all baselines benefited highly from globalization. This is the reason that PACFL and IFCA could achieve a high performance with only 2 clusters.

In order to better assess the potential of the SOTA baselines under a real-world and challenging Non-IID task where the local data of clients have strong statistical heterogeneity, we design the following experiment naming it as MIX-4. We assume that each client owns data samples from one of the four datasets, i.e., USPS (Hull 1994), CIFAR-10, SVHN, and FMNIST. In particular, we distribute CIFAR-10, SVHN, FMNIST, USPS among 31, 25, 27, 14 clients respectively (100 total clients) where each client receives 500 samples from all classes of only one of these dataset. This is a hard Non-IID task. We compare our approach with the SOTA baselines in the classification of these four different datasets, and we present the average of the clients' final local test accuracy in Table 3. As can be seen, IFCA is unable to effectively handle this difficult scenario with tremendous data heterogeneity with just two clusters, as suggested in (Ghosh et al. 2020) as the best fitting number of clusters. IFCA (2) with 2 clusters performs almost as poorly as the global baselines while PACFL can find the optimal number of clusters in this task (four clusters) and outperforms all SOTA by a large margin. The results of IFCA (4) with 4 clusters is 76.79 ± 0.43. As observed, PACFL surpasses all the global competitors (by $+14\%, +15\%, +16\%, +8\%$ for FedAvg, FedProx, FedNova, SCAFFOLD) as well as all the personalized competitors (by $+19\%, +35\%, +7\%, +16\%$ for LG, PerFedAvg, IFCA, CFL, respectively). Further, the visualization in Fig. 3c and 3d also show how PACFL determines the optimal number of clusters on MIX-4.

How Many Clusters Are Needed?

As we emphasized in prior sections, one of the significant contributions of PACFL is that the server can easily determine the best fitting number of clusters just by analyzing the proximity matrix without running the whole federation. For instance, for IID scenarios, we expect the best fitting number of clusters to be one. The reason behind is that under IID setting, since all clients have similar distributions, they can share in the training of the averaged global model to benefit all. On the other hand, in the case of MIX-4, we expect the best fitting number of clusters to be four. More generally, we expect the best fitting number of clusters to be dependent on the similarities/dissimilarities in distributions among the



(a) **(b)** **(c)** **(d)**

Figure 3: The main message of this visualization is to understand the cluster structure of different datasets as well as the distribution similarity of different heterogeneous tasks/datasets. (a) depicts the UMAP visualization of CIFAR-10 classes. As can be seen, CIFAR-10 naturally has two super clusters, namely animals (cat, dog, bird, deer, horse, frog) and vehicles (car, plane, ship, truck), which are shown in the purple and green regions, respectively. This means that within each super cluster, the distance between the distribution of the classes is small. While the distance between the distributions of the two super clusters are quite huge. Since the union of clients data is CIFAR-10, two cluster is enough to handle the Non-IIDness across clients. This is the reason that the best accuracy performance on CIFAR-10 is obtained when the number of clusters is 2. (b) We obtained the proximity matrix $\mathbf{A}$ as in Eq. 2 and sketched it. The entries of $\mathbf{A}$ are the smallest principle angle between all pairs of classes of CIFAR-10. This concurs with (a) showing the cluster structure of CIFAR-10. (c) The data of MIX-4 is naturally clustered into four clusters. The structure of the 4 clusters is also accurately suggested by PACFL for this task. (d) We did the same thing as in (b) for MIX-4 as well and sketched the matrix.

Algorithm	MIX-4	Algorithm	MIX-4
SOLO	55.08 ± 0.29	LG	58.49 ± 0.46
FedAvg	63.68 ± 1.64	PerFedAvg	42.60 ± 0.60
FedProx	61.86 ± 3.73	IFCA (2)	70.32 ± 3.57
FedNova	60.92 ± 3.60	CFL	61.18 ± 2.63
Scaffold	69.26 ± 0.84	**PACFL**	$\mathbf{77.83 \pm 0.33}$

Table 3: The benefits of PACFL are particularly pronounced when the tasks are extremely Non-IID. This table evaluates different FL approaches in the challenging scenario of MIX-4 in terms of the top-1 test accuracy performance. While all competing approaches have substantial difficulties in handling this scenario with tremendous data heterogeneity, the results clearly show that PACFL is very robust even under such difficult data heterogeneity scenarios.

clients. We empirically show in Fig. 2 that the best accuracy results on CIFAR-100, CIFAR-10, SVHN, and FMNIST for Non-IID label skew (20%) are obtained when the number of clusters are 2, 2, 2, and 4, respectively.

The UMAP (McInnes, Healy, and Melville 2018) visualization in Fig. 3a also confirms that two clusters is the best case for training the local models on partitions of CIFAR-10 dataset. Broadly speaking, CIFAR-10 has 2 big classes, i.e., class of animals (cat, dog, deer, frog, horse, bird) and class of vehicles (airplane, automobile, ship and truck). Fig. 3b also depicts the proximity matrix of CIFAR-10 dataset, whose entries are the principal angle between the subspace of every pairs of 10 classes (labels). This further confirms that our proposed method perfectly captures the level of heterogeneity, thereby finding the best fitting number of clusters in a privacy preserving manner. In particular, our experiments demonstrate that the clients that have the sub-classes of these two big classes have common features and can improve the performance of other clients that own sub-classes of the same big class if they are assigned to the same cluster. A similar observation can be seen for other datasets.

In PACFL, the server only requires to receive the *signature* of the clients data in one-shot and thereby initiating the federation with the best fitting number of clusters. This translates to several orders of magnitude in communication savings for PACFL. However, as mentioned in (Ghosh et al. 2020), IFCA treats the number of clusters as a hyperparameter which is optimized after running the whole federation with different number of clusters which increases the communication cost by several orders of magnitude.

Generalization to Newcomers

PACFL provides an elegant approach to handle *newcomers* arriving after the federation procedure, to learn their personalized model. In general, for all other baselines it is not clear how they can be extended to handle clients unseen at training (federation). We show in Algorithm 3 how PACFL can simply be generalized to handle clients arriving after the end of federation, to learn their personalized model. The unseen client will send the signature of its data to the server and the server determines which cluster it belongs to. The server then sends the corresponding model to the newcomer and the newcomer fine tunes the received model. To evaluate the quality of newcomers' personalized models, we design an experiment under Non-IID label skew (20%) where only 80 out of 100 clients are participating in a federation with 50 rounds. Then, the remaining 20 clients join the network at the end of federation and receive the model from the server and personalize it for only 5 epochs. The average final local test accuracy of the unseen clients is reported in Table 4. Focusing on CIFAR-100, as observed, some of the personalized baselines including LG and PerFedAvg perform as poor as global baselines and SOLO. PACFL consistently outperforms other baselines by a large margin.

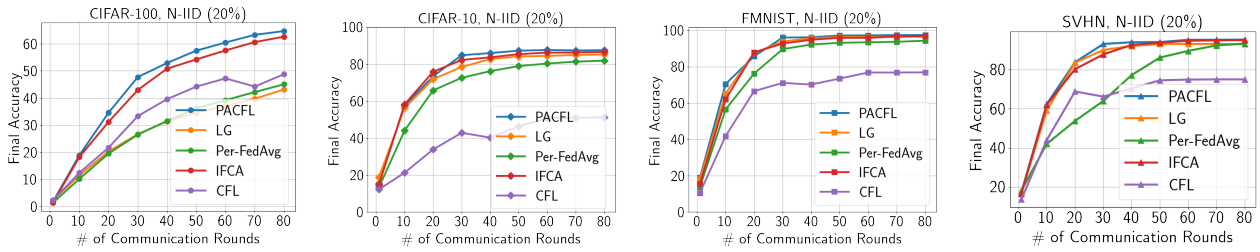


Figure 4: Test accuracy versus number of communication rounds for Non-IID (20%). PACFL converges fast to the desired accuracy and consistently outperforms strong competitors.

Algorithm 3: Generalization to newcomers after federation

1: **Server:** An existing $\mathbf{A}$, $\mathbf{U}$, $\{C_1, ..., C_Z\}$, clustering threshold β
2: **Require:** a set of B newcomers and their corresponding first p significant singular vectors, i.e. $\mathbf{U}_{new} = [U_p^1, ..., U_p^B]$
3: $\mathbf{A}$, $\mathbf{U}$ = PME($\mathbf{A}$, $\mathbf{U}$, $\mathbf{U}_{new}$) % Alg. 2 %
4: $\{C_1, ..., C_Z\} \leftarrow$ HC($\mathbf{A}, \beta$) % Updating the clusters and determining the cluster ID of each new client k %
5: Each new client k receives the corresponding cluster model $\theta_{g,z}$ from the server
6: FineTune($k; \theta_{g,z}$): by SGD training % this step is optional %

Algorithm	FMNIST	CIFAR-10	CIFAR-100	SVHN
SOLO	95.13±0.42	82.30±1.00	27.26±0.98	91.5 ± 0.64
FedAvg	77.61±3.78	31.01±1.83	32.19±0.32	71.78±3.43
FedProx	74.30±4.70	27.56±3.24	32.41±1.17	74.30±4.70
FedNova	74.66±2.81	31.48±1.49	33.18±0.80	73.04±3.65
Scafold	73.97±1.68	37.22±1.34	23.90±2.61	64.96±4.74
LG	94.58±0.33	77.98±1.61	10.63±0.21	89.48±0.65
PerFedAvg	89.88±0.38	73.79±0.51	30.09±0.35	67.48±2.88
IFCA	96.29±0.04	84.98±0.41	55.66±0.20	94.83±0.14
PACFL	**96.36 ± 0.20**	**87.14 ± 0.15**	**59.16 ± 0.42**	**95.25 ± 0.08**

Table 4: Average local test accuracy across unseen clients on different datasets for Non-IID label skew (20%).

Communication Cost

Learning with Limited Communication In this section we consider circumstances that frequently arise in practice, where a limited amount of communication round is permissible for federation under a heterogeneous setup. To this end, we compare the performance of the proposed method with the rest of SOTA. Herein, we consider a limited communication rounds budget of 80 for all personalized baselines and present the average of final local test accuracy over all clients versus number of communication rounds for Non-IID label skew (20%) in Fig. 4. Our proposed method requires only 30 communication rounds to converge in CIFAR-10, SVHN, and FMNIST datasets. CFL yields the worst performance on all benchmarks across all datasets, except for CIFAR-100. Per-Fedavg seems to benefit more from higher communication rounds. IFCA, is the closest line to ours for CIFAR-10, SVHN and FMNIST, however, PACFL consistently outperforms IFCA. This can be explaiend by the fact that IFCA randomly initializes cluster models that are inherently noisy, and many rounds of federation is required until the formation of clusters is stabilized. Further, IFCA is sensitive to initialization and a good initialization of cluster model parameters is key for convergence (Rad, Abdizadeh, and Szabó 2021). This issue can be further pronounced by the results presented in Table 5 which demonstrate the number of communication round required to achieve a designated target accuracy. In this table, "−−" means that the baseline is unable to reach the specified target accuracy. As can be seen, PACFL beats IFCA and all SOTA methods.

Algorithm	FMNIST	CIFAR-10	CIFAR-100	SVHN
Target	75	80	50	75
FedAvg	200	−−	130	150
FedProx	200	−−	115	200
FedNova	−−	−−	120	150
Scafold	−−	−−	−−	−−
LG	13	33	−−	16
PerFedAvg	19	60	110	39
IFCA	14	25	40	17
CFL	47	−−	−−	−−
PACFL	**12**	**24**	**37**	**15**

Table 5: Comparing different FL approaches for Non-IID (20%) in terms of the required number of communication rounds to reach target top-1 average local test accuracy.

Conclusion

In this paper, we proposed a new framework for clustered FL that directly aims to efficiently identify distribution similarities among clients by analyzing the principal angles between the client data subspaces spanned by their principal vectors. This approach provides a simple, but yet effective clustered FL framework that addresses a broad range of data heterogeneity issues beyond simpler forms of Non-IIDness like label skews.

References

Bonawitz, K. A.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 1175–1191. ACM.

Day, W. H.; and Edelsbrunner, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1): 7–24.

Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive Personalized Federated Learning. *CoRR*, abs/2003.13461.

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

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An Efficient Framework for Clustered Federated Learning. In *Advances in Neural Information Processing Systems 33*.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773.

Haddadpour, F.; and Mahdavi, M. 2019. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*.

Hallac, D.; Leskovec, J.; and Boyd, S. P. 2015. Network Lasso: Clustering and Optimization in Large Graphs. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, 387–396. ACM.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778. IEEE Computer Society.

Hershey, J. R.; and Olsen, P. A. 2007. Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, IV–317–IV–320.

Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5): 550–554.

Jain, P.; Netrapalli, P.; and Sanghavi, S. 2013. Low-rank matrix completion using alternating minimization. In Boneh, D.; Roughgarden, T.; and Feigenbaum, J., eds., *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, 665–674. ACM.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119, 5132–5143. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Toronto*.

LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4): 541–551.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021a. Federated Learning on Non-IID Data Silos: An Experimental Study. *CoRR*, abs/2102.02079.

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

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, March 2-4, 2020*. mlsys.org.

Liang, P. P.; Liu, T.; Ziyin, L.; Salakhutdinov, R.; and Morency, L.-P. 2020a. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Liang, P. P.; Liu, T.; Ziyin, L.; Salakhutdinov, R.; and Morency, L.-P. 2020b. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three Approaches for Personalization with Applications to Federated Learning. *CoRR*, abs/2002.10619.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

McMahan, B.; and Ramage, D. 2017. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3.

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

Morafah, M.; Vahidian, S.; Wang, W.; and Lin, B. 2022. FLIS: Clustered Federated Learning via Inference Similarity for Non-IID Data Distribution. In *NeurIPS Workshop*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Qian, G.; Sural, S.; Gu, Y.; and Pramanik, S. 2004. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM symposium on Applied computing*, 1232–1237.

Rad, H. J.; Abdizadeh, M.; and Szabó, A. 2021. Federated Learning with Taskonomy for Non-IID Data. *CoRR*, abs/2103.15947.

Sarcheshmehpour, Y.; Leinonen, M.; and Jung, A. 2021. Federated Learning from Big Data Over Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, 3055–3059. IEEE.

Sattler, F.; Müller, K.; and Samek, W. 2021. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Trans. Neural Networks Learn. Syst.*, 32(8): 3710–3722.

Talwalkar, A.; Kumar, S.; Mohri, M.; and Rowley, H. 2013. Large-scale SVD and manifold learning. *The Journal of Machine Learning Research*, 14(1): 3129–3152.

Vahidian, S.; Morafah, M.; Chen, C.; Shah, M.; and Lin, B. 2022. Rethinking Data Heterogeneity in Federated Learning: Introducing a New Notion and Standard Benchmarks. In *NeurIPS 2022 workshop*.

Vahidian, S.; Morafah, M.; and Lin, B. 2021. Personalized federated learning by structured and unstructured pruning under data heterogeneity. *IEEE ICDCS*.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Advances in Neural Information Processing Systems*, volume 33, 7611–7623. Curran Associates, Inc.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.